2014-06

Improving the prototyping process in Department of Defense acquisition

Coble, Mar

Monterey, California: Naval Postgraduate School

http://hdl.handle.net/10945/42718

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

SYSTEMS ENGINEERING CAPSTONE PROJECT REPORT

IMPROVING THE PROTOTYPING PROCESS IN DEPARTMENT OF DEFENSE ACQUISITION

by

Team BlackberryPI
Cohort 311-124G

June 2014

Project Advisors: Brigitte T. Kwinn
Bonnie Young

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.			
1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** June 2014	**3. REPORT TYPE AND DATES COVERED** Capstone Project Report	
4. TITLE AND SUBTITLE IMPROVING THE PROTOTYPING PROCESS IN DEPARTMENT OF DEFENSE ACQUISITION			**5. FUNDING NUMBERS**
6. AUTHOR(S) Cohort 311-124G / Team BlackberryPI			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol number ____N/A____.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

The current Department of Defense (DOD) multiyear acquisition process is too costly and takes far too long for weapon systems to be developed. To help the DOD address this challenge a model was developed that will mature and transition technology into formal system development. The team utilized a tailored systems engineering strategy, including requirements analysis, functional architecting, modeling, simulation, and risk analysis when developing the Technology Development System (TDS) model. The TDS is based on risk assessment, detailed planning, and early system prototyping in order to successfully proceed into formal system development with proven technologies. This model was developed with the intent that it be extendable to all program offices within the DOD. The TDS leveraged attributes and known best practices from doctrinal sources combined into a step-by-step development process. The context surrounding successful prototyping still lacks the proper knowledge-based approach needed to make the effort worthwhile. The architecture, model, and simulation together provide the traceability, validation, and system requirements to define system entry criteria, accurately plan and conduct technology maturation, and reduce the cost and technical risk associated with early system development within the DOD acquisition life cycle.

14. SUBJECT TERMS: Prototyping, Acquisition, Systems Engineering, Technology risk reduction			**15. NUMBER OF PAGES** 265
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2–89)
Prescribed by ANSI Std. 239–18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

IMPROVING THE PROTOTYPING PROCESS IN DEPARTMENT OF DEFENSE ACQUISITION

Cohort 311-124G/Team BlackberryPI

Mark Coble
Jeremy Royster

Doug Glandon
John Stewart

Phi Pham
Brandon Taylor

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN SYSTEMS ENGINEERING

AND

David Bailey   Keith Herndon

MASTER OF SCIENCE IN ENGINEERING SYSTEMS

from the

NAVAL POSTGRADUATE SCHOOL
June 2014

Lead editor: John Stewart

Reviewed by:

Brigitte T. Kwinn
Project Advisor

Bonnie Young
Project Advisor

Accepted by:
Cliff Whitcomb
Systems Engineering Department

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The current Department of Defense (DOD) multiyear acquisition process is too costly and takes far too long for weapon systems to be developed. To help the DOD address this challenge a model was developed that will mature and transition technology into formal system development. The team utilized a tailored systems engineering strategy, including requirements analysis, functional architecting, modeling, simulation, and risk analysis when developing the Technology Development System (TDS) model. The TDS is based on risk assessment, detailed planning, and early system prototyping in order to successfully proceed into formal system development with proven technologies. This model was developed with the intent that it be extendable to all program offices within the DOD. The TDS leveraged attributes and known best practices from doctrinal sources combined into a step-by-step development process. The context surrounding successful prototyping still lacks the proper knowledge-based approach needed to make the effort worthwhile. The architecture, model, and simulation together provide the traceability, validation, and system requirements to define system entry criteria, accurately plan and conduct technology maturation, and reduce the cost and technical risk associated with early system development within the DOD acquisition life cycle.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

AMRDEC	Aviation and Missile Research, Development, and Engineering Center
AoA	Analysis of Alternatives
ASD(R&E)	Assistant Secretary of Defense for Research and Engineering
BCL	Business Capability Life cycle
CDD	Capability Development Document
CDR	Critical Design Review
COCOM	Combatant Command
CONOP	Concept of Operation
COR	Contracting Officer Representative
COTS	Commercial-Off-the-Shelf
CPD	Capability Production Document
DAG	Defense Acquisition Guidebook
DARPA	Defense Advanced Research Projects Agency
DAU	Defense Acquisition University
DPM	Deputy Project Manager
DOD	Department of Defense
DODAF	Department of Defense Architecture Framework
DODI	Interim Department of Defense Instruction
DT&E	Developmental Test and Evaluation
EAR	Entrance Acceptance Review
EFFBD	Enhanced Functional Flow Block Diagram
EM	Evaluation Measures
EMD	Engineering and Manufacturing Development
FCS	Future Combat System
FY	Fiscal Year
GAO	Government Accountability Office
GMR	Ground Mobile Radio
GOGO	Government Owned Government Operated

ICOM	Input, Control, Output and Mechanism
ICD	Initial Capabilities Document
I/O	Input/Output
IOT&E	Initial Operational Test and Evaluation
IDEF0	Integrated Definition 0
IMS	Integrated Master Schedule
INCOSE	International Council on Systems Engineering
IPR	Integrated Project Review
IPT	Integrated Project Team
IRB	Institutional Review Board
JCIDS	Joint Capabilities Integration and Development System
JROC	Joint Requirements Oversight Council
JTRS	Joint Tactical Radio System
KPP	Key Performance Parameters
KSA	Key Systems Attribute
LIB	Less is better
LFT&E	Live Fire Test and Evaluation
MBSE	Model-Based Systems Engineering
MCAP	Manned/Unmanned Common Architecture
MIB	More Is better
MDA	Milestone Decision Authority
MDAP	Major Defense Acquisition Program
MDD	Materiel Development Decision
MS	Milestone (Milestone A, B, C, etc)
MSA	Materiel Solution Analysis
MSSE	Master’s of Science in Systems Engineering
MSES	Master’s of Science in Engineering Systems
NASA	National Aeronautics and Space Administration
NPS	Naval Postgraduate School
NSC	National Security Council
OV-1	Operational View – 1
ORS	Operationally Responsive Space

OT&E	Operational Test and Evaluation
OUSD (AT&L)	Office of the Under Secretary of Defense for Acquisition, Technology and Logistics
P&D	Production and Deployment
PDR	Preliminary Design Review
PIF	Prototype Integration Facility
PM	Project Manager
POTUS	President of the United States
PWS	Performance Work Statement
RA	Requirements Analysis
RDECOM	Research, Development and Engineering Command
RDT&E	Research, Development, Test and Evaluation
RM	Risk Manager
RMP	Risk Management Plan
SADT	Structured Analysis and Design Technique
SE	Systems Engineering
SEDP	System Engineering Design Process
SIMILAR	State Investigate Model Integrate Launch Assess Re-evaluate
SME	Subject Matter Expert
SOS	System of Systems
SPEC	Systems and Proposal Engineering Company
TD	Technology Development
TDS	Technology Development System
T&E	Test and Evaluation
TMRR	Technology Maturation and Risk Reduction
TPM	Technical Performance Measures
TRA	Technology Readiness Assessment
TRL	Technology Readiness Level
TTP	Technology Transition Plan
UAV	Unmanned Aerial Vehicle
WBS	Work Breakdown Structure
WSARA	Weapon Systems Acquisition Reform Act

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

The Department of Defense (DOD) relies on its Program Executive Offices, Program Managers, Science and Technology directorates, and industry partners to develop innovative technologies for weapon systems. Several GAO, RAND, and independent reports have addressed problems in early system prototyping, as well as the ability to transition technologies into formal system acquisition programs. (GAO 2006) The purpose of this Naval Postgraduate School Capstone Project was to develop a process based method that would enable the successful maturation and transition of technology to formal system acquisition at Milestone (MS) B. This process, titled Technology Development System (TDS) is based upon solution-neutral modeling and simulation. The results of the models, simulations, and viewpoints provide a proof of concept that will meet the DOD's need to develop and provide relevant weapon system capabilities to the warfighter more quickly. The capstone team recommends that the DOD implement this technology development model prior to formal system acquisition at MS B. Implementation of the TDS during the Technology Maturation and Risk Reduction phase of acquisition will enhance the DOD's ability to assess, develop, and transition matured capabilities into system acquisition.

The primary objective of this capstone effort was to produce an extensible technology assessment and development method that can be applied to technologies that have previously achieved a technology readiness level (TRL) 4 prior to entering the TDS process. The TDS process and model:

1) Define a standard technology assessment method in order to accurately and objectively determine the strengths and weaknesses of an incoming technology.

2) Identify the appropriate planning structure to develop an accurate and feasible programmatic and technical plan for maturing and transitioning the technology.

3) Introduce the opportunity within the structure of the model to either redefine the development plan or terminate the development effort should it become necessary.

Due to academic time constraints, the capstone team was directed, consistent with the priorities of the Naval Postgraduate School (NPS), toward a specific phase of the DOD Acquisition life cycle. This pilot process model only focuses on the methods necessary to assess a technology entering MS A, develop the technology, and transition the successfully demonstrated technology into formal system acquisition at MS B. The challenges were documenting the overarching definition of prototype and prototyping as it relates to the DOD as a whole, identifying the current prototyping environment and how it is used to reduce technical risk, identifying the appropriate set of activities needed to successfully develop and transition technology, and compiling the root causes of weapon system development failures and the creating a process model that overcomes these gaps in technology development.

Executable models of functional activities were developed to simulate the operation of the TDS functions and to measure its performance. The resultant data was used to validate the development process using historical systems development programs. Use cases, based on documented DOD acquisition programs, were researched and relevant acquisition data, namely cost, schedule, and performance, was used as supporting validation parameters for the TDS model. The model was validated and verified with multiple use case scenarios designed with expected outcomes and actual use case outcomes. The intent behind the model was not to show that the TDS approach was the only solution to system development but instead was a proof of concept. The simulation demonstrated, that if properly implemented and executed, the TDS approach provides increased opportunities to track maturation.

There are a multitude of relevant systems engineering models that have been tailored to meet specific needs. The capstone team developed a tailored systems engineering process based on Bahill and Gissing's SIMILAR process (Bahill and Gissing 1998). The SIMILAR process was adapted according to the system of interest, activities to be completed, and the NPS Capstone environment. Tailoring of the full SE process was performed to scale the rigorous application of the SIMILAR processes to an appropriate level based on need and the system development context (INCOSE 2010).

Based upon research and stakeholder analysis, the team determined that the DOD needed a standardized and tailorable prototyping process that provided organized principles, synergistic programmatic and technical methodologies, and success metrics in order to support effective early acquisition prototyping and technological development. In response to this need, a solution-neutral, process model was developed to assess the technical and programmatic feasibility of developing the technology, along with the planning, iterative technical reviews, and transition strategies to ensure successful future development once the system entered formal acquisition at MS B. The model was developed with a multidisciplinary team using Systems Engineering (SE) processes learned throughout the course of study.

The following key deliverables were created for NPS and the Research, Development and Engineering Command (RDECOM):

- Repeatable and extensible process model for addressing the DOD early system prototyping challenges
- Innoslate Model executable reference architecture with bi-directional links between system entities and attributes
- Innoslate executable simulation model
- Draft Model Taxonomy for the Technology Development System
- Resource Data

The TDS provides an extendable DOD technology maturation and transition process, specifically within the Technology Maturation and Risk Reduction (TMRR) Phase, to facilitate delivery of a flexible solution to the user that meets mission needs. The TDS model and its associated activities will satisfy the DOD’s need to comprehensively and objectively assess program feasibility, plan for technological development, mature the technology, transition the technology into formal system development, while also providing the necessary iterative loop that allows a redefinition or termination of the effort should it be deemed appropriate. The TDS proof of concept was developed as a model based, solution-neutral process meant to provide an extendable technology development model for the Technology Maturation and Risk Reduction phase of acquisition.

List of References

Bahill, A T, and B Gissing. 1998. “Re-Evaluating Systems Engineering Concepts Using Systems Thinking.” *IEEE Transaction on Systems, Man and Cybernetics, Part C: Applications and Reviews.* Vol. 28, No. 4. 516–527.

GAO. 2006. Best Practices: Stronger Practices Needed to Improve DOD Technology Transition Processes. Washington, D.C.: GAO.

INCOSE. 2010. Systems Engineering Handbook : A Guide For System Life Cycle Processes And Activities. San Diego: INCOSE, SE Handbook Working Group.

ACKNOWLEDGMENTS

Team BlackberryPI would like to thank their thesis advisors, professors Brigitte T. Kwinn and Bonnie Young, for their continuous support, guidance and insight throughout this capstone project. Further, the team would like to acknowledge all the professors, classmates, and co-workers who have provided input into all aspects of this report. Finally, to the families of the team: our deepest gratitude and heartfelt thanks for their love, patience and support throughout the entire process.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

The current Department of Defense (DOD) multiyear acquisition process is too costly and takes far too long for weapon systems to be developed. Because of rapid changes in the threat, mission, and technological environments, a system may be ineffective in meeting mission needs or be deemed obsolete once it is fielded (Erwin 2013). To solve this problem, the impact of technology development and prototyping processes were analyzed.

The inability of DOD programs to sufficiently reduce technology risk prior to entering formal systems development has, over the past five years, contributed to a 13% cost growth in weapon systems acquisition and a 17% increase in schedule for the initial operational capability (Copeland et al. 2013). The Defense Acquisition Guidebook (DAG) and DOD 5000 series documents have undergone numerous revisions to narrow the focus of Technology Development in an effort to achieve true risk reduction in weapon systems acquisition. Similar themes have resulted from the numerous GAO, RAND, and independent research reports (Drezner and Huang 2009; GAO 2006; GAO 2012):

- Technology Maturity is a major indicator of design complexity, adequate requirements, and program risk
- System prototype demonstrations play a vital role in achieving a successful technology development strategy
- Competitive early systems prototyping can provide an added benefit through the incentive of competition

The acquisition environment of today, with further shrinking budget and schedule allocations, requires effective systems engineering in order to meet the warfighter's needs. Prior to formal systems development (Pre-Engineering and Manufacturing Development [EMD]) prototyping can provide a significant advantage in reducing technology risk. The research, analysis, and development performed in support of this capstone project is intended to provide a foundation for the systems engineering and prototyping activities necessary for successful technology maturation and transition during the Pre-EMD phase of acquisition.

A. PROJECT OVERVIEW

This capstone project was initiated as a result of the issues highlighted in numerous studies specific to prototyping facilities within the DOD. These studies have identified the need for an improved technology development and early system prototyping process in DOD acquisition. The goal was to identify the specific activities, processes, and sequencing that should be performed during the Technology Development phase of the DOD acquisition life cycle in order to ensure successful system development. To accomplish this task, a systems engineering approach was applied to current prototyping practices used in the DOD to identify potential methods to improve the process.

The first step was to define the problem, which included stakeholder analysis, operational concept development, system context identification, and value system modeling for the team's Technology Development System (TDS) concept. Functional analysis and decomposition enabled the definition of the TDS in terms of its functional architecture and how it transforms system inputs into system outputs. Functional decomposition, in the sense of systems engineering, views the functions and their interfaces as building blocks for the system. Parsing functions with their associated performance, quality, physical, informational and other views further improved the ability to characterize the system (Blanchard and Fabrycky 2011). Developing and iterating the functional architecture of the TDS promoted the precise definition and structure of the relationships between the whole and its parts.

Applying a system engineering approach enabled robust problem solving techniques while also providing an assurance that all likely aspects of the system were considered and integrated into the whole. The following activities represent just a portion of the major activities performed during the course of executing the tailored systems engineering process:

- Requirements Analysis – encompasses the activities that went into determining the needs of the system and the various stakeholder needs. The requirements were analyzed for traceability, testability, measurability.
- Use Cases – conveys, along with the associated metrics, how the system should interact with the user to achieve the stated system goal. The use cases describe the ways in which the system is intended to be used and to show the steps needed to perform system tasks.
- Functional Models – provides a graphical representation of the system that describes the functions and their associated decisions, actions, and activities that connect the system functions together. Developing the functional architecture models for the TDS allowed the team to graphically describe the dynamic process of the system.
- Simulation – validates the interaction between the system functions. Executable simulation was performed in order to move beyond the models that provide a description of what the system is "supposed to do" and provide a representation or prediction of what the system "will do." The simulation results provided a data set that enabled the team to validate predicted outcomes of the system and formulate recommendations for system implementation. (3SL 2014).

B. TEAM OVERVIEW

A team of students from the Naval Postgraduate School (NPS) in the Master's of Science in Systems Engineering (MSSE) and Master's of Science in Engineering Systems (MSES) Distance Learning Cohort 311–124G were selected for this project. The team members, as presented in Table 1, are employees of the Aviation and Missile Research, Development, and Engineering Center (AMRDEC) of the RDECOM.

Table 1. Team Members

Team Member	Organization
David Bailey	AMRDEC Technical Management Directorate, Matrixed to Lower Tier Project Office
Mark Coble	AMRDEC Aviation Engineering Directorate, Utility Helicopter Division
Doug Glandon	AMRDEC Technical Management Directorate, Matrixed to Cargo Helicopter Program Management Office
Keith Herndon	AMRDEC Aviation Engineering Directorate, Cargo Helicopter Division
Phi Pham	AMRDEC Software Engineering Directorate, Aviation Division
Jeremy Royster	AMRDEC Aviation Engineering Directorate, Structures and Materials Division
John Stewart	AMRDEC Aviation Engineering Directorate, Special Operations Aircraft Division
Brandon Taylor	AMRDEC Aviation Engineering Directorate, Utility Helicopter Division

Detailed information regarding the roles and responsibilities for the team members are in the project plan included in the Appendix of this report.

C. PROJECT BACKGROUND

According to Erwin, the current DOD multiyear acquisition process can make the development and procurement of weapon systems a slow and arduous task. The process begins with the task of trying to predict future capabilities. Next, the requirements to support these capabilities must be generated. After the requirements have been documented, an exhaustive design and development effort begins which includes multiple reviews, milestones, tests, and evaluations of the system being developed. The system is not delivered to the user until all of these activities have been successfully completed (DOD 2013; Erwin 2013).

In many cases, the process can take several years or even decades before a system is delivered to the users. An example is the Air Force F-22 Raptor, which took over two decades to reach the field. Another example of the inefficiency during system development and acquisition is the F-35 Joint Strike Fighter program. During prolonged development, as seen with the F-22 and the F-35, the threat environment can change drastically, which can result in a system being ineffective in meeting new mission needs (Erwin 2013).

Another concern has been the low return on DOD technological investments in recent years. In the article by Erwin (2013), "Defense Technologists Advocate 'Early Prototyping' of Future Weapons," Defense Secretary Chuck Hagel was quoted as saying that weapon systems are "taking longer, costing more, and delivering less than initially planned and promised." This, in part, has been due to ineffective DOD prototype development of its systems. In the past ten years alone, $50 billion was spent on developing systems that have not materialized (Erwin 2013).

Even with the current issues in the utilization of prototyping, defense industry experts, as well as, the Under Secretary of Defense agree, that effective prototyping along with changes in procurement regulations would provide a rapid acquisition environment that could help to offset many of the DOD weapon system development problems (Erwin 2013). These views are substantiated by the mandates stated in the Weapon Systems Acquisition Reform Act of 2009 (WSARA). The WSARA requires the use of prototyping earlier in the system life-cycle process. The difficulty with early prototyping (prior to

Milestone B) is determining the appropriate level of design, construction, and capabilities needed to represent and evaluate a system. At this stage, if prototype development is not selective, its cost may exceed its benefits (Borowski 2012).

The WSARA was not the first recommendation to use prototyping to improve the DOD acquisition process. The Packard Commission, also known as the Presidents Blue Ribbon Commission on Defense Management, suggested, “a high priority should be given to building and testing prototype systems and subsystems before proceeding with full-scale development.” The commission also asserted that the focus on prototyping would allow the DOD to “know how much it will cost before we buy (Packard 1986).” In spite of these recommendations, many DOD acquisition programs still experience significant technical, cost, and schedule problems with respect to system development and fielding (Borowski 2012).

The issues discussed in the previous section have also had an adverse effect on the technological development of DOD weapon systems. In a 2007 Government Accountability Office (GAO) annual assessment, only 16% of DOD programs had successfully reached a mature technology level at Milestone B. Of the programs that had reached the Critical Design Review (CDR), only 44% had attained technological maturity and 24% exhibited stable designs. Of the programs that had transitioned to MS C, only 67% possessed a mature technology and one third still had unstable designs. Even with these percentages, 47% of the programs were moving forward with production prototype development (Gordon 2008). This data helps to explain how technological development and prototyping of DOD weapon systems has been inefficient over the past decade.

D. PROJECT ASSUMPTIONS

The team used experience and research to develop assumptions. The list evolved throughout the project as information and facts were established. The following assumptions assisted in bounding the scope of the project and guiding the systems engineering analysis.

- In some cases, the prototype process may not be the most feasible solution. If prototype development exceeds the expected life-cycle benefits, other alternatives may need to be considered (GAO 2012).
- In most projects, an efficient prototyping process will result in lower cost, schedule, or technical risks (Erwin 2013).
- Prototyping will continue to be one of the essential steps in the development of the majority of DOD weapon systems.
- There are no statutory, regulatory, and certification requirements for this project.
- The Technology Readiness Level (TRL) definitions and descriptions, as stated in this report, will not change.
- Changes to the DOD's current prototyping processes will be considered a valid recommendation.
- The TDS will be working with immature technologies with TRLs of at least 4.
- The TDS must work within the current acquisition system. Funding for the TDS will be provided with each technology maturation candidate program/project.
- The customer will provide initial requirements for the expected capability of the matured technology.
- A primitive need is provided for the technology entering the TDS.
- A program schedule is provided for the technology entering the TDS.
- The need for a prototype has been confirmed.

These assumptions reduced the gap between known conditions, facts, and unknown information regarding the development of the system.

E. PROJECT CONSTRAINTS

The team identified the project constraints during project definition and planning using research and guidance provided by the advisors. The following constraints convey the project limitations and restrictions, which also serve to bound the scope of the project and guide the systems engineering analysis.

- The research, analysis, and recommendations associated with this capstone project must be completed by June 2014 (3 academic quarters).
- The project must be accomplished by the project team (8 members).
- The team is limited to 10 man hours of project work per week.
- The team's initial knowledge of the project subject is limited. Research and analysis will be conducted to overcome this constraint.
- The project must be accomplished without incurring any monetary costs beyond that which is already been invested in the MSSE and MSES program tuition.
- The project must be accomplished using only unclassified, open source information.
- Project academic deliverables must meet NPS guidelines.
- The project must comply with the NPS human research protection program Institutional Review Board (IRB) Student Research guidelines for human subject research
- The deliverables will be produced using standard programs and formats (Microsoft Excel, Word, PowerPoint, etc.).
- Funding follows a five year cycle, therefore the next opportunity to fund the system will be in five years.
- The U.S. government has a limited budget for changes to the acquisition process. Changes to the acquisition process must consider the expense given the budget constraints.

F. LITERATURE REVIEW AND RESEARCH QUESTIONS

Thorough research of academic, governmental, and industry sources provided pertinent information to establish the background, problem statement, and develop system requirements. In addition, the team researched and analyzed multiple systems engineering models to provide a better understanding of the systems engineering processes required for effective prototyping.

1. Research Questions

The following questions guided this research:

- How is prototyping defined with respect to DOD acquisition?
- How does the DOD currently use the prototyping process to reduce technical risk?
- Why is early acquisition prototyping not currently realizing success in the DOD acquisition process?
- What activities are performed in early acquisition prototyping?
- What metrics can measure prototyping success?

2. Problem Background and Definition Research

The genesis of the project came from an article by Sandra Erwin (2013) entitled "Defense Technologists Advocate 'Early Prototyping' of Future Weapons." The article documents comments made by Charles Hagel, United States Secretary of Defense, about DOD acquisitions, "taking longer, costing more, and delivering less than initially planned and promised." Erwin discusses how prototyping is not being used effectively and offers examples of prototype failures and successes within the DOD. This information will be utilized to identify areas in the prototyping process that have the potential to be improved (Erwin 2013).

The problem statement for this project was influenced by the paper written by Mark Borowski (2012) titled "Competitive Prototyping in the Department of Defense: Suggestions for a Better Approach." Borowski explores the history of prototyping within the DOD and presents the lessons learned through prototyping. His paper also validates some of the alternatives considered by the team and offers helpful insights into the benefits and shortfalls of prototyping. Borowski also discusses the need to define the terms "prototype" and "prototyping" and offers possible definitions.

Along with Borowski's paper, the report from the Packard Commission (1986) also offered insight into the history of prototyping in the world of DOD acquisition. The Packard Commission (1986) specifically recommends prototyping as a method of better management within the DOD, because it was one of six factors found in many successful commercial programs. This report provided credence to the value of prototype development, as well as, a checklist for analyzing the completeness of current DOD prototyping activities (Packard 1986).

To understand successful examples of DOD prototyping, a technical report from the Systems Engineering Research Center was studied. The report by Fracktor and Colombi (2012) titled "Expedited Systems Engineering for Rapid Capability and Urgent Needs," studied systems engineering in rapid prototyping. The DOD operates several prototyping centers and program offices within the military services who specialize in rapid prototyping, such as the Prototype Integration Facility (PIF) at Redstone Arsenal, the Big Safari Air Force Program Office at Wright-Patterson Air Force Base, and the Operationally Responsive Space (ORS) Office at Kirtland Air Force Base (Facktor and Colombi 2012). This report studied these and many other government and industry organizations to determine what makes these successful in rapidly producing prototypes that successfully meet the needs of the warfighters. The team utilized the content of this report to determine metrics to be applied during value system design development.

Another source, published by the DOD (2012), titled the *Defense Acquisition Guidebook (DAG*), was studied to understand the current DOD prototyping process. The DAG describes the prototyping phase, its purpose and its expected outputs and outcomes. Currently, the DOD sees the prototyping phase as a way to determine cost-effective designs and perform risk reduction activities (DOD 2012). Reviewing the DAG provided insight into how prototyping fits within the larger DOD acquisition process and areas where the team can focus its analysis efforts and recommendations for improvement (DOD 2012).

The Interim Department of Defense Instruction (DODI) 5000.02, which was released in November of 2013, also provided additional insight into the application of prototyping in DOD acquisition. It states that hardware prototypes may govern the schedule, decision points, and milestones of a program, but software development will dictate its progression. Therefore, software development must be closely coordinated and integrated with hardware prototype development. In addition, it states that competitive prototyping may be required as a Milestone A activity. Further, risk reduction prototypes may be required to reduce engineering and manufacturing development risks during the TMRR Phase (DOD 2013).

The Interim DODI 5000.02 also addressed some changes in the utilization of prototypes from its 2008 release. When feasible, prototyping will be implemented into Operational Test & Evaluation (OT&E) activities. In most cases, high cost weapon systems, such as spacecraft and naval vessels, will not require a production prototype as part of MS C (DOD 2013). These changes provided valuable input into the development of the project's system.

Dr. Judith Dahmann's (2010) presentation titled, "Early Systems Engineering" provided insight into how the DOD has been trying to improve early systems engineering with the release of DODI 5000.2 and the WSARA of 2009. The WSARA provides specific language that defines how prototyping is to be utilized in weapon system development prior to Milestone B (Dahmann 2010; Borowski 2012).

To provide the team with a practical understanding of prototyping, Todd Z. Warfel's (2009) *Prototyping: A Practitioner's Guide* was reviewed. This guide identified processes and principles to be utilized during prototype development. It also described the benefits and value of prototyping (Warfel 2009).

3. Systems Engineering Process Research

Blanchard and Fabrycky's (2011) *Systems Engineering and Analysis* provided the team with an introduction to systems engineering as a life-cycle approach to system development. The authors describe methodologies, concepts, tools, and models that can be applied iteratively to system acquisition, beginning with the identification of a need and culminating in the delivery, operation, support, and phase-out of a system (Blanchard and Fabrycky 2011). An understanding of these principles was paramount in the application of a systems engineering approach that provides effective improvements for early system prototyping and technological development of DOD weapon systems.

The source utilized for the project's systems engineering process was tailored using the "State, Investigate, Model, Integrate, Launch, Assess and Re-evaluate"

(SIMILAR) process developed by Bahill and Gissing (1998). This model was selected because it was easily adaptable and encompassed the essential steps required for effective system development.

4. Literature Review Conclusion

The literature review aided in identifying background, problem statement, and develops system requirements. It was also useful in identifying additional questions about prototyping that needed to be addressed. The team’s research also identified examples of successful prototyping that could be useful to the project. Other sources provided insight into the current DOD prototyping process and how it is trying to improve early systems engineering through prototyping. The team also reviewed literary sources that increased their knowledge of the prototyping process and to assist in the development of a tailored systems engineering process for the project.

G. PROBLEM STATEMENT

In order to understand the problem and develop the problem statement, the team was provided with a general project description (Erwin 2013). Based on this information and additional research, the team developed the following problem statement:

The current DOD multiyear acquisition process is too costly and takes far too long for weapon systems to be developed. Because of rapid changes in the threat, mission, and technological environments, a system may be ineffective in meeting mission needs or be deemed obsolete once it is fielded (Erwin 2013). Recent acquisition reform has made prototyping early in system development a requirement. This is a step in the right direction; however, the prototyping methods being utilized have not been as effective as anticipated in providing the information needed to guide future decisions and minimize risk (Borowski 2012).

After establishing the problem statement, several systems engineering models were reviewed and analyzed to identify an approach that was applicable to resolving the problem. From this activity, a tailored process was developed to guide the team through the problem solving process.

H. SYSTEMS ENGINEERING (SE) PROCESS

The technical approach for this project was defined using the SE process depicted in Figure 1. This process was adapted and tailored from the "State, Investigate, Model, Integrate, Launch, Assess and Re-evaluate" (SIMILAR) process as stated by Bahill and Gissing (1998). The Bahill and Gissing (1998) process includes seven tasks: "State the problem, Investigate alternatives, Model the system, Integrate, Launch the system, Assess performance, and Re-evaluate" The authors of the model also state that these tasks should be performed concurrently rather than sequentially (Bahill and Gissing 1998).

The derived systems engineering structure embodies the recursive and iterative nature of systems engineering while also aligning with NPS and RDECOM deliverables. The tailored process includes specific activities to be performed during the problem refinement, use case development and system modeling, simulation, and recommendation phases. Based on "Systems Engineering and Analysis" (Blanchard and Fabrycky 2011) and the needs statement, the project resulted in the creation of the functional baseline finally culminating in a recommended method to assess technical and programmatic feasibility upon system entry, effectively plan the technology development process, perform early system prototyping, and successfully transition a technology into formal system development. The team will apply the principles attained during the NPS Systems Engineering master's program in the execution of these activities.

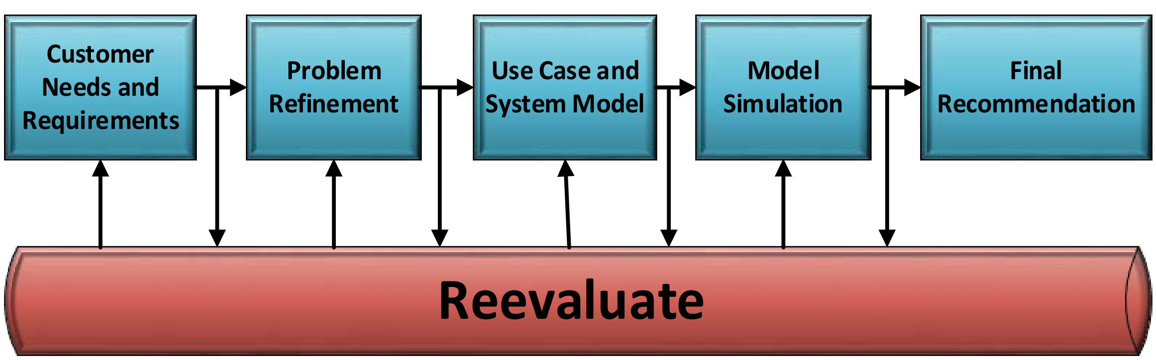


Figure 1. Tailored Systems Engineering Process (after Bahill and Gissing, 1998)

1. Needs and Requirements and Problem Refinement Phase

The SE process began with the customer needs and problem refinement phase as shown in Figure 1. The problem definition ended with an understanding of the effective need, a refined problem statement, and an agreed upon scope limiting the analyses and system development to the Technology Development phase of the DOD Acquisition life cycle, and the proposed path forward.

The needs analysis process began with the initial problem statement. The needs analysis was conducted in a systematic manner in order to identify gaps between the current state and the desired state. The difference between the current and desired state was analyzed in an iterative manner to appropriately identify the need. This research and analysis allowed the team to systematically bound the concerns that had to be addressed for effective technology assessment, planning, maturation, and early system prototyping in the Technology Development phase of the DOD Acquisition life cycle.

Requirements and Stakeholder analysis was conducted using the background information collected during research. The system requirements were derived from the needs and were used as inputs for system design and development. Stakeholder analysis was performed to identify the groups that affect or are affected by the proposed system under development.

At this point in the systems engineering process, a need was identified and transformed into a set of systems requirements. The next logical step in the process is to develop an architecture that will serve as the foundation for system development. The functional architecture was created, refined, and iterated throughout the systems engineering process.

The value system design exercise established a qualitative value model that identified the functions, sub-functions, and evaluation measures. It also served to identify system characteristics and measurable parameters for use in system modeling and simulation.

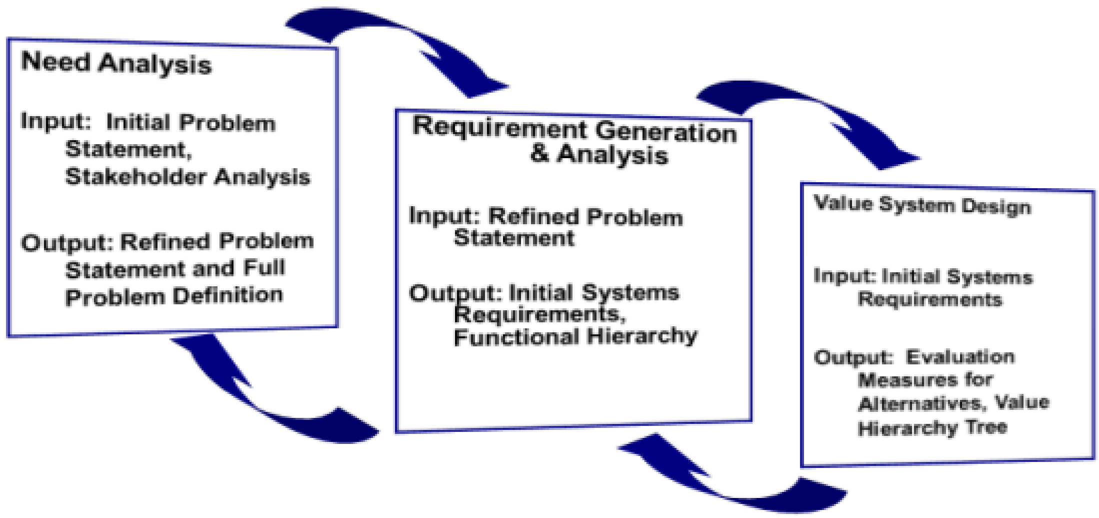


Figure 2. Needs and Requirements and Problem Refinement Phases (after Acosta et al. 2007)

2. Use Case and System Model Phase

Model development enabled the team to clarify and describe system behavior. The models help to decompose the system, as a whole, into smaller and more easily definable blocks that can be analyzed from the perspective of functional flows, inputs and outputs, resources, and the physical relationships. By decomposing the larger system into separate elements, the team gained insight and understanding into how to organize the system into an effective construct. The model based systems engineering (MBSE) approach was important because it provided a consistent framework around which to construct the parts of the system along with their attributes and relationships in order to produce a complete representation of the system (Scott 2011).

The team tailored the Design Alternatives Phase of Acosta et al. (2007) to develop the Use Case and System Model phase. During this phase, as depicted in Figure 3, the value hierarchy and functional analysis provided a basis for the selection of a use case and served as an input for modeling the system. Parameters associated with the system's top-level sub-functions were selected to identify the type of use case data needed for input into the system model. The use cases identify the ways in which the system is exercised by the users. Sequence diagrams, typically Enhanced Functional Flow Block Diagrams

(EFFBD), graphically illustrate the interactions the system executes in support of the desired outputs.

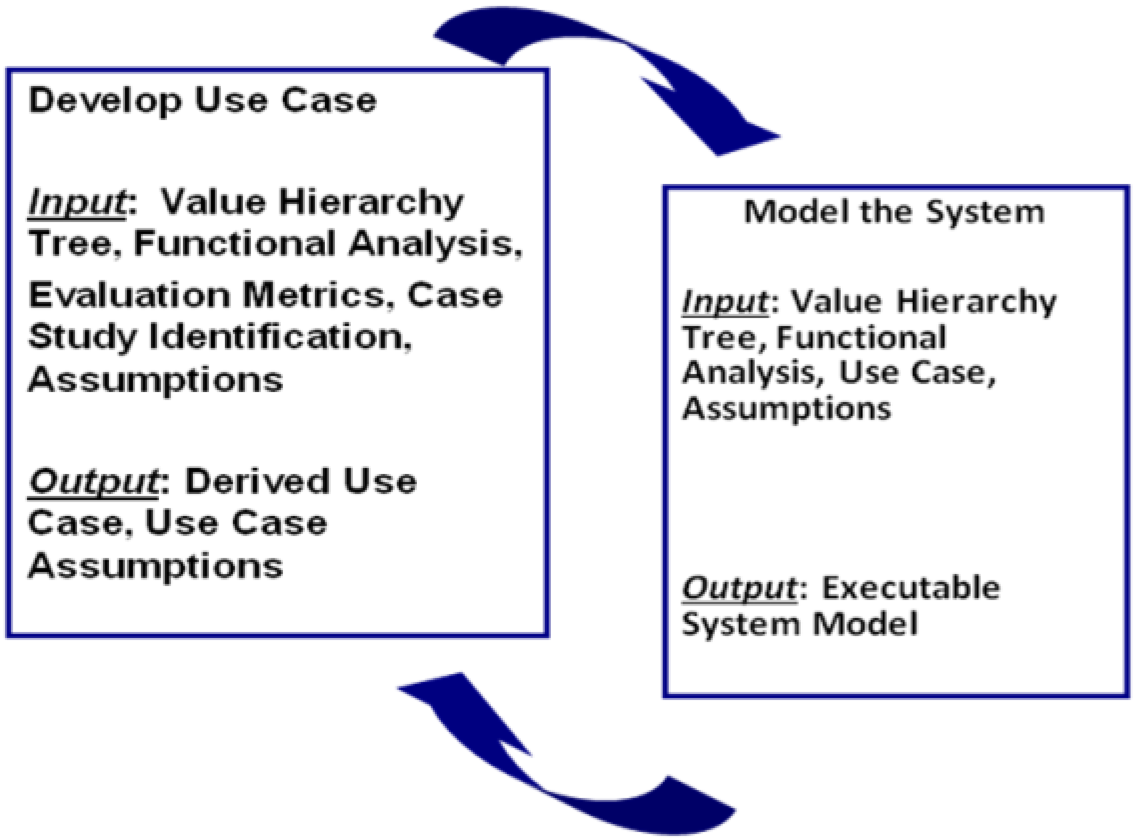


Figure 3. Use Case and System Model Phase (after Acosta et al. 2007)

3. Model Simulation Phase

The project team tailored Acosta et al.'s (2007) Simulation and Analysis of Alternatives (AoA) Phase to develop a simulation phase used to validate the TDS model. The framework for this phase is presented in Figure 4.

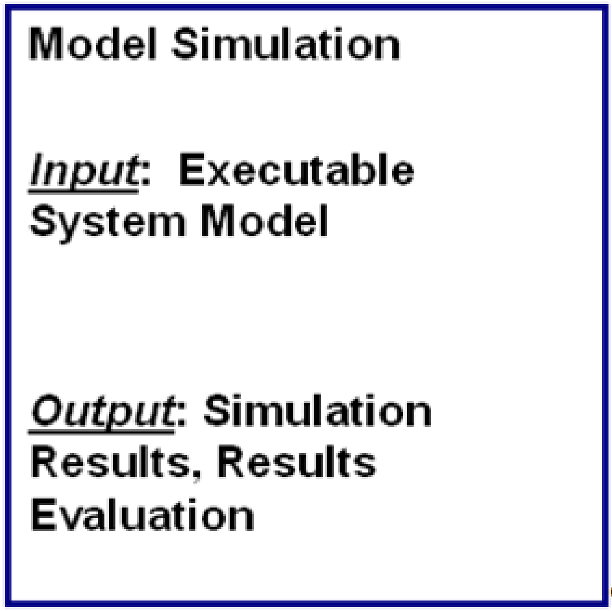


Figure 4. Model Simulation Phase (after Acosta et al. 2007)

Simulation modeling tools are available commercially that offer the powerful capability to simulate a wide variety of events. When selecting the simulation tool to apply to this project, it was important to the team that the simulation tool was robust enough to allow for the recreation of events defined in the system functional architecture and correlate those events to the specific use cases to be simulated. Innoslate, provided commercially by Systems and Proposal Engineering Company (SPEC) Innovations, offered the discrete event simulation capability that allowed the team to execute the system models. Control parameters, developed as part of the use case scenarios, were required to represent an operational scenario. The parameters act as triggers that keep the simulation moving in the appropriate pattern throughout the specified steps. The results from the simulation provided a means for validating the performance of the system with respect to the selected functional parameters. The resultant data produced from the simulations also provided confidence in the system architecture model and became the foundation for the Final Recommendation Phase.

4. Final Recommendation Phase

The result of executing the systems engineering activities was to identify and select recommendations to be provided to the stakeholders. These recommendations were based on the evaluated results analyzed throughout the SE process. The team leveraged the findings produced by the extensive research and evaluation to develop the TDS. It was imperative to refine the strengths and weaknesses of the initial findings in order to develop the final solution. The simulation portion of the SE process served to validate the conclusions and make the proper recommendations to improve the effectiveness of planning technology development in the Pre-EMD phase of acquisition, mature the technology through early system prototyping, and successfully transition the technologies of DOD weapon systems.

I. SYSTEM LIFE CYCLE

The life-cycle phases for the TDS are depicted in Figure 5. These phases were tailored from the system life-cycle model as depicted in Blanchard and Fabrycky's (2011, 30) S*ystems Engineering and Analysis*.

The first phase, definition of need, focuses on the customer's needs and requirements, analysis of the need, and problem refinement. Use case development, modeling and simulation, and a final recommendation are the activities included in the design and development phase. In the implementation phase, the selected solution is transformed into an actual system. The utilization phase takes the developed system and puts it into use. The last phase optimizes and improves the TDS, as needed, to increase its effectiveness, as well as, extend its life cycle. This phase also addresses the retirement of the system (Blanchard and Fabrycky 2011).

Figure 6 correlates the system life-cycle phases with the tailored systems engineering process. This figure serves to correlate the tailored systems engineering process with the TDS life-cycle model.

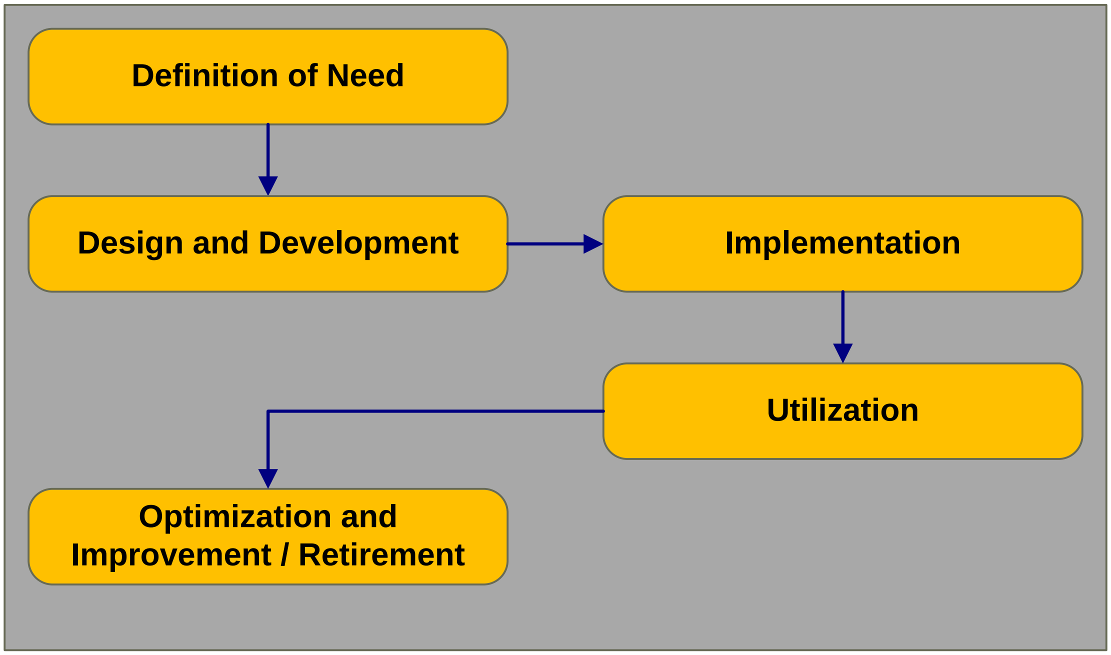


Figure 5. Tailored System Life-cycle Phases (after Blanchard and Fabrycky 2011, 30)

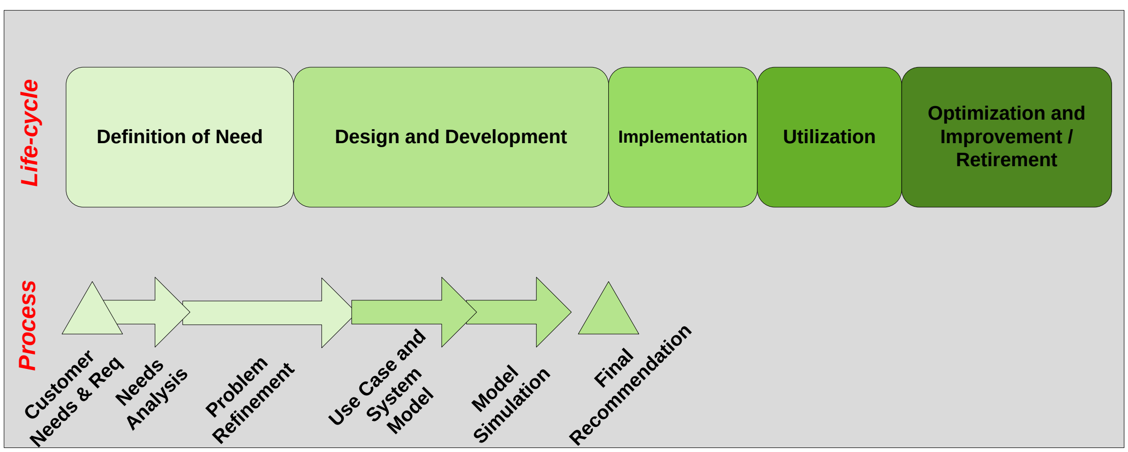


Figure 6. Life-cycle Phases and SE Process Correlation

J. PROJECT RISK

During the capstone project, the team performed risk management and analysis for the system process under development. The Risk Management Plan (RMP) was developed using DOD's (2006) *Risk Management Guide for DOD Acquisition*. Each team member has a responsibility to ensure risks are identified, analyzed, tracked and mitigated to ensure that the project will remain on track to meet its goals. Risk management efforts began during the needs and requirements and problem refinement phases and all technical and non-technical risks were identified and documented. These risks were modified, supplemented and closed as the project progressed through each phase.

The status of current project risks and their associated mitigation and contingency plans were documented in a risk management database and briefed to the team and NPS advisors on a bi-weekly basis. The risk management plan is included in Appendix A and contains the processes and methods for the team's approach to project risk management.

The team performed a risk analysis for the system utilization and retirement phases. The risk analysis will determine the current and future prototyping risks. This risk analysis will be provided later in the report after the system has been designed, evaluated and analyzed for areas of concern with regard to developing effective prototypes.

K. TEAM ORGANIZATION AND STRUCTURE

The team organizational structure provided a means to coordinate the activities of the project to best accomplishes the project objectives. In addition, it was utilized to coordinate team functions to facilitate communication and interaction among the people involved in the project. The organizational structure also established the roles, relationships, authority and responsibility for ensuring the success of the program.

The team organization included the students from Redstone Arsenal participating in the NPS 311–124G cohort. The project management responsibilities for this capstone project include the Project Manager (PM), Deputy Project Manager (DPM), Scheduler, Action Officer, Configuration Manager, Editor-in-Chief, and Risk Manager. The teams also organized into two smaller integrated product teams (IPTs) for focus and to execute the project technical approach. The organizational roles are shown in Figure 7 and IPTs in Figure 8.

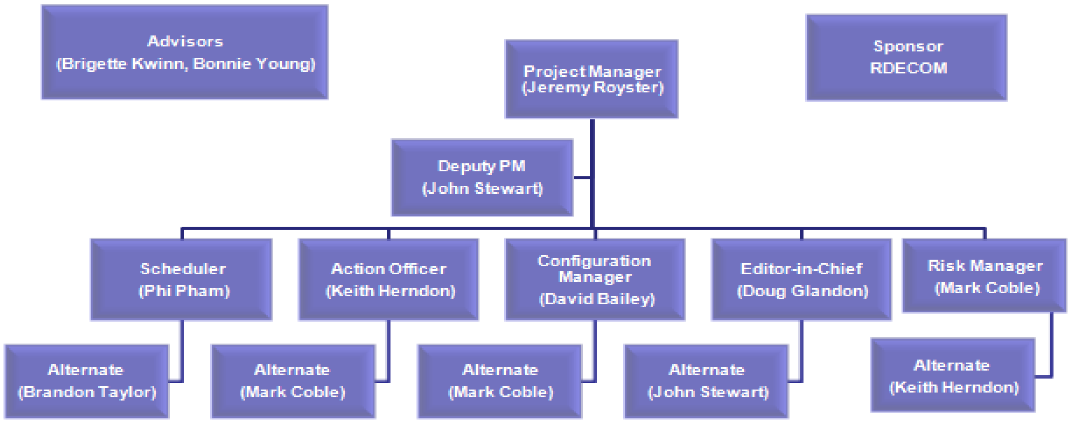


Figure 7. Team Technical Management Organization

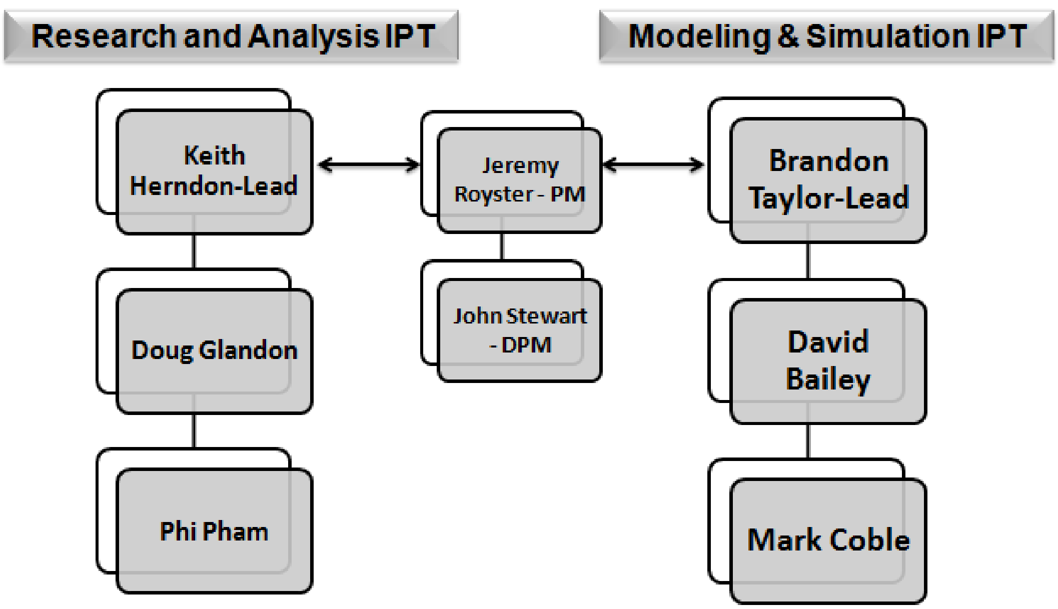


Figure 8. Technical IPTs

The organization of the IPTs in Figure 8 is in response to the activities performed in the tailored SE process. The Research and Analysis IPT was focused on the research and analysis portions of our SE process. The literature review, research questions, current DOD Acquisition process, review, and case studies were all evaluated by the Research and Analysis IPT. The Modeling and Simulation IPT was focused on creating the system concept model, the current DOD Acquisition model and simulating those models with data from the use case development. The PM and DPM were setup to manage the project actions and workloads for each of the IPTs along with assisting in technical reviews and actions for both IPTs.

L. SUMMARY

Chapter I provided a discussion of the project background, assumptions, and constraints to scope the problem. The research questions guided and focused the literature review, which was used to develop the problem statement and guide the system development. The technical approach for this project was defined and depicted in the tailored SE process and correlated to the TDS prototype development system life cycle.

The risk management plan and team organizational structure were identified to clarify the execution of the project's systems engineering process. Development of the systems engineering process along with iterative refinement of the problem statement was key to developing system requirements. The system requirements, operational concept definition, and stakeholder needs analysis will be described and discussed in Chapter II.

II. DEFINITION OF NEEDS, REQUIREMENTS, AND PROBLEM REFINEMENT

The first step in the SE process was the identification of the problem in the Needs, Requirements and Problem Refinement phase. It is essential that the systems engineering process begin by defining the problem and its importance. Defining the problem proved to be one of the most difficult parts of the process, particularly with the time constraints inherent in a capstone setting. There were several false starts that resulted in schedule delays before the team could lay an adequate foundation from which to progress. The capstone team took an iterative approach to define the problem in qualitative and quantitative terms in enough detail to justify progressing to the next step (Blanchard and Fabrycky 2011). After defining the problem completely, the team proceeded into the needs analysis. The goal of the needs analysis was to translate the broadly defined “want” into a more specific system-level requirement (Blanchard and Fabrycky 2011). Identifying the problem and accomplishing the needs analysis took the ultimate team approach to ensure that the “whats” were identified prior to developing the “hows” of the system.

A. PROBLEM STATEMENT

In order to understand the problem and develop the problem statement, the team was provided with a general project description and problem statement presentation. Based on this information and additional research, discussed in the Literature Review and Research section in Chapter I.F, the team developed the following problem statement:

> The current DOD multiyear acquisition process is too costly and takes far too long for weapon systems to be developed. Because of rapid changes in the threat, mission, and technological environments, a system may be ineffective in meeting mission needs or be deemed obsolete once it is fielded. (Erwin 2013)

Recent acquisition reform has made prototyping early in system development a requirement. This is a step in the right direction; however, the prototyping methods being utilized have not been as effective as anticipated in providing the information needed to guide future decisions and minimize risk (Borowski 2012). Even though early system prototyping has been recognized, required, and considered a best practice in a multitude of sponsored and independent reports, the DOD still delegates the responsibility for the prototyping process, as well as the decision as to whether prototyping is needed, down to the program manager level. This has resulted in ad hoc prototyping occurrences and disparate methodologies among the military branch acquisition constructs. There is no formal prototyping process model that has been accepted by the program offices within the DOD.

After establishing the problem statement, the team reviewed the DOD acquisition process and identified the key stakeholders. The stakeholders are the groups affected by and invested in problem outcome (Erwin 2013).

B. STAKEHOLDER ANALYSIS

The system stakeholder list, as presented in Table 2, is a top-level view of the entities that affect or will be affected by the proposed actions of the project. The desires of the stakeholders were developed from the research conducted on the Chapter I Literature Review section reference material and the role the individual stakeholder has in the DOD acquisition technology development process. This information was utilized to determine the specific goals and needs of the stakeholders. It was also crucial for conducting the needs analysis to determine the system requirements and priorities. Through this analysis, the system requirements were linked directly to the stakeholder’s needs.

Table 2. Stakeholder Goals and Needs

Stakeholder	Active/Passive	Goals/Needs
Warfighter	Active	The warfighter needs relevant weapon systems that fully support mission requirements
Secretary of Defense	Active	The SECDEF needs relevant weapon systems developed and delivered within cost and schedule to support DoD initiatives
Under Secretary of Defense	Active	Provides procurement guidance Oversees the buying of all equipment Use prototyping as a way to keep engineering community employed during upcoming defense budget reduction
Technical Development Centers (TDCs)/Laboratories	Active	Design and develop prototype systems
Deputy Assistant Secretary for Army Research &Technology	Active	Use prototyping to better understand and evaluate requirements and what is achievable
Deputy Assistant Secretary for Navy Research & Technology	Active	Use prototyping to better understand and evaluate requirements and what is achievable
Deputy Assistant Secretary for Air Force Research & Technology	Active	Use prototyping to better understand and evaluate requirements and what is achievable
Defense Advanced Research Projects Agency (DARPA)	Active	Evaluate current and future weapon system technologies and capabilities
Defense Contractors	Active	Support the Defense TDCs/Laboratories. Goal: Manufacture or produce a product that satisfies the customer. Need: Make profit and the most efficient manufacturable and producable product.
US Taxpayers	Passive	Goal: Aid the warfighters in safely doing their job with a cost effective soution. Need: Need to avoid major tax increases while also providing useful tools/products to our warfighters
Academia	Passive	Need to review and understand the SE processes necessary to improve the rapid prototyping acquistion process.

From the stakeholder goals and needs the team derived a basic list of consolidated and common “wants” for the system solution. The stakeholders want to:

- Develop effective systems
- Develop Relevant systems
- Develop Cost effective systems
- Develop systems in a timely manner
- Support DOD initiatives
- Improve prototyping procurement guidance
- Improve prototyping viability

- Use prototyping to refine system requirements
- Evaluate future technologies
- Increase production efficiency
- Understand prototyping SE process

The list of "wants" were important for continuing to refine the problem. The team's next step was to perform a needs analysis to translate the "wants" into refined needs and initial system requirements (Blanchard and Fabrycky 2011).

C. NEEDS ANALYSIS

A needs analysis, by its very nature, identifies and characterizes gaps in existing capabilities that serve as roadblocks to achieving a specified goal. The needs analysis is a large part of the initial steps toward new developments or process improvements. The challenge for the capstone team was to clarify the needs of the stakeholders in unambiguous terms and ensure the needs were supported and documented by research and analysis.

1. Problem Importance

As previously discussed in the Background Section of this report, the DOD's track record for delivering relevant weapon systems to the Warfighter, within budget and on schedule, in recent years has been poor. GAO assessments have affirmed an alarming trend in cost growth and schedule slips of weapon system programs. The focal point of this problem was attributed to the lack of knowledge needed to attain a successful design at key acquisition points (Gordon 2008). In addition, this problem has been compounded with reductions in the defense budget (Borowski 2012).

Prototyping was identified as one of the methods that could reverse this trend; however, a process is needed that can assist the program manager and the developer in determining the proper level of prototype development (Gordon 2008). In most cases, prototyping an entire system is cost prohibitive (Borowski 2012); therefore, a process is needed that focuses on selecting capabilities to be prototyped that exhibit the highest level of technical risk to the system. This process would serve to reduce development cost and risk, while providing an opportunity to resolve problems in the early stages of the acquisition life cycle (Borowski 2012). Implementation of this process would improve the value of prototyping by providing the knowledge needed to achieve a successful design at key decision points in system development. Numerous GAO reports have concluded that most DOD programs proceed with low a level of technology knowledge resulting in cost/schedule increases (GAO 2006; GAO 2008; GAO 2012). Only 16% of programs achieved mature technology at MS B. Programs that did not have mature technologies upon entry into MS B averaged 32% cost growth and a 20 month schedule delay (Gordon 2008). Throughout the capstone process, intensive research into successful and unsuccessful DOD weapon system programs has revealed a common theme: knowledge supersedes risk over time. Positive acquisition outcomes require knowledge based approaches before any significant development commitments are made. The successful programs have anchored their approach in attaining and demonstrating technical knowledge at critical decision points in the process (GAO 2013).

2. Need Statement

One approach to demonstrate a technological capability that is prevalent within the DOD acquisition framework is the use of prototyping. The blanket statement, "we can use prototyping," however, is not as simple as it sounds. There are many layers, both technical and programmatic, that form the context for prototyping a particular technology or capability. One key aspect of realizing successful demonstration through prototyping is to have a common understanding among all vested program stakeholders as to what is meant by the terms "prototype" and "prototyping." The DOD has been performing some level of prototyping for fifty years or more, however, the definitions for these terms remain varied and undefined (Borowski 2012). A set of standardized definitions for the terms "prototyping" and "prototype" and their different applications must be developed. These definitions and applications need to be acceptable to the defense acquisition community for the purpose of providing guidance in the development and assessment of technologies and capabilities. For example, a prototype may be viewed as a test article that is utilized to demonstrate a technology or the capabilities of a system. When the test article is actually "tested," it may be viewed as the "act of prototyping." (Borowski September 2012) Standardizing these terms and their applications could improve the effectiveness of prototype development.

The team utilized stakeholder analysis, along with the systems engineering analysis, to refine the initial problem statement into an effective needs statement. The development of the needs statement was a result of researching the challenges plaguing the DOD in the areas of technology maturity, systems engineering, early system prototyping, and knowledge gaps at key stages of acquisition. This research became the baseline for scoping the problem and subsequent need. An iterative feedback loop during problem definition and needs analysis ensured that the problem was properly identified and the systems engineering process would be appropriate for solving the problem.

The revised and final Effective Needs Statement for this capstone project is as follows:

> The DOD needs to change their current multiyear acquisition process in order to develop and provide weapon system capabilities to the Warfighter more quickly and support the Warfighter's need to adapt to rapid changes in threat, mission, and technological environments, within the constraints of controlling and/or reducing costs given fiscal instability, and providing solutions that are relevant and delivered in a timely manner. (Erwin, 2013)

3. System-Level Functional Requirements

The system level requirements were based on the established need discussed in the preceding paragraphs. These requirements were developed to ensure that the operational need for the system would be satisfied. System-level functional requirements describe "what" functions the system must perform in order to meet stakeholder needs. Each functional requirement is described in the following bullets:

- The system shall assess program feasibility. The purpose of assessing program feasibility is to objectively and rationally determine the strengths and weakness of a technology. This functional requirement also prescribes that the system evaluates programmatic elements such as its developmental strategies, risks, cost, and schedule.

- The system shall produce a plan for technological development. Technological development entails planning the programmatic and technical work required to transition a technology. To accomplish this, the system must determine a spend rate for funding, manpower requirements, and plan for hardware and software development and integration.
- The system shall mature the technology. This functional requirement ensures the system executes the development plan for maturing a technology.
- The system shall provide the capability to redefine the maturation program planning. Redefinition is a key part of any technological development effort. This functional requirement serves to provide the option to revise planning if the original maturation plan is proven inadequate to meet the desired result.
- The system shall provide the capability to terminate the maturation of the technology. Numerous doctrinal sources cite the inability or unwillingness of managers to terminate a failing system development effort. This functional requirement provides the option to terminate the maturation of the technology due to undesired result during maturation or due to customer requirement or need changes.
- The system shall transition the technology. This functional requirement facilitates the transition of the technology back to the Program Office of Record.
- The system shall close out the matured technology. This functional requirement facilitates the closing of the maturation and transition process by providing the response to the service request. The service response is generated for terminated and transitioned technologies.

The needs analysis was completed resulting in a set of system-level functional requirements. Establishing these requirements was an important step in transitioning to the design and development phases of the system. The next step was to develop an operational concept for the system that would meet the functional requirements. The functional analysis and functional architecture modeling can be found in Chapter III of this report.

D. TECHNICAL APPROACH

The technical approach for this capstone project was to create a technology development process for use during the TMRR Phase of the DOD Acquisition life cycle. Reports have consistently revealed that technological maturity and early systems prototyping performed prior to formal systems acquisition at MS B is a primary driver for technology risk reduction during formal system acquisition (GAO 2013). The development and execution of any technology development or early system prototyping process cannot be conceived or completed in a vacuum. There are many factors that contribute to successful prototyping. Consideration must be given to the acquisition phase entry point, the initial maturity of the technology to be demonstrated, and the intended outcome of the prototyping activities. The TDS was developed to synergize these associated activities, both technical and programmatic, in a manner consistent with the tenets of systems engineering in order to satisfy the DOD's effective need.

E. OPERATIONAL CONCEPT DEFINITION

During the extensive research for problem definition, it became clear there was opportunity for improvement in the DOD acquisition process. Due to academic constraints, it became necessary to bound TDS development to specifically the Technology Maturation and Risk Reduction phase of acquisition. This phase occurs after a successful MS A decision and is focused on identification, development, and transition of technologies or capabilities into formal systems acquisition at MS B. The TDS concept was developed within this acquisition life-cycle boundary as well as the constraints and assumptions listed in Table 3.

Table 3. TDS Constraints and Assumptions

TDS Constraints	TDS Assumptions
The TDS must operate within the confines of the current DOD acquisition structure.	The TDS will focus on prototyping early in the acquisition process, between MS A and MS B.
The TDS must accept technology of a TRL ≥ 4 and mature it to a TRL of 6.	The TDS will serve as an opportunity to mature promising technology and capabilities.
The TDS must have review points to ensure the technology is maturing according to the plan and on budget.	The TDS will serve as an opportunity to reject those technologies and capabilities that cannot meet the needed TRL.
A primitive need is provided for the technology entering the TDS.	The TDS process can be performed by government organizations or contractor organizations
A program schedule is provided for the technology entering the TDS.	The need for a prototype to advance the development of the technology has been confirmed.

As an integral part of the operational concept development, an Operational Concept (OV-1, Figure 9) was developed for the TDS. The OV-1 is part of the Department of Defense Architecture Framework (DODAF) and is a tool for developing and documenting architectures. Within the DODAF, the OV-1 provides a graphical depiction of the high level interactions between the system and its environment. (DOD CIO 2010)

Figure 9 is a visual representation of the operational concept and is assembled to be viewed from left to right. From the left, a combination of capabilities, needs, and requirements generated by the Warfighter, Combatant Command (COCOM) , National Security Council (NSC), Congress, or the President, as indicated by the green lines, are passed to the DOD for evaluation and consideration (TRADOC 2011) The Defense Advanced Research Projects Agency (DARPA) also provides input into this process by presenting new technologies, concepts, and processes to the DOD and into the DOD

Acquisition Process, as indicated by the blue lines (DARPA 2013) The black two-way arrow represents the flow of capabilities, requirements, directives, and funding between the DOD and into the DOD Acquisition Process. The red lines denote the products, requirements, directives, and funding flows into the DOD Acquisition process from either government owned or contractor owned facilities. The output of the DOD Acquisition Process, which are physical products, are produced and delivered to a customer as depicted by the purple lines.

Within the DOD Acquisition Process icon is a separate entity that represents the DOD Acquisition Prototyping Process. This separation was the result of research identifying "trade space" between these two processes where prototyping and technology development could be improved (AMRDEC 2014).

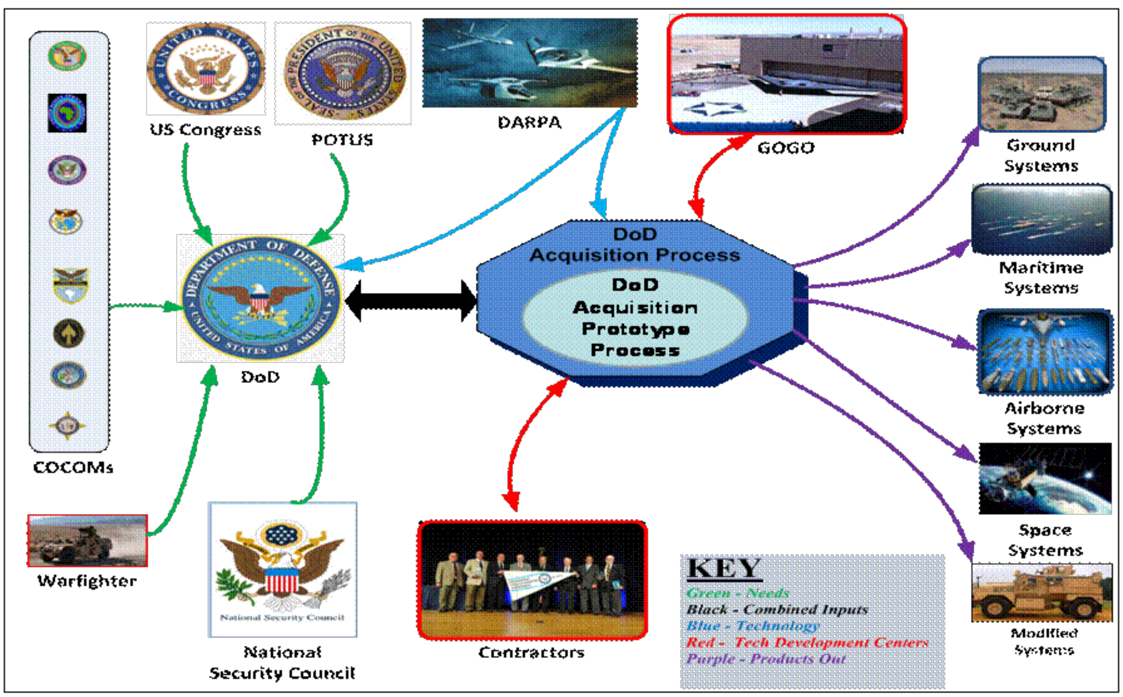


Figure 9. Operational View (TDS OV-1)

Focus areas for the system will cover perspectives inherent to the user, the program manager, and the science and technology communities. Current doctrine and processes form the basis of the TDS and provide the necessary building blocks for the solutions that are developed as part of the system and implemented within the DOD acquisition framework.

The TDS will provide new and/or improved acquisition processes, specifically within the Technology Maturation and Risk Reduction Phase, to facilitate delivery of a flexible solution to the user that meets mission needs. It will also provide value-added capability with a high probability of technological success to program management offices, enable identification and achievement of performance, and reduce technological risk to the science and technology community.

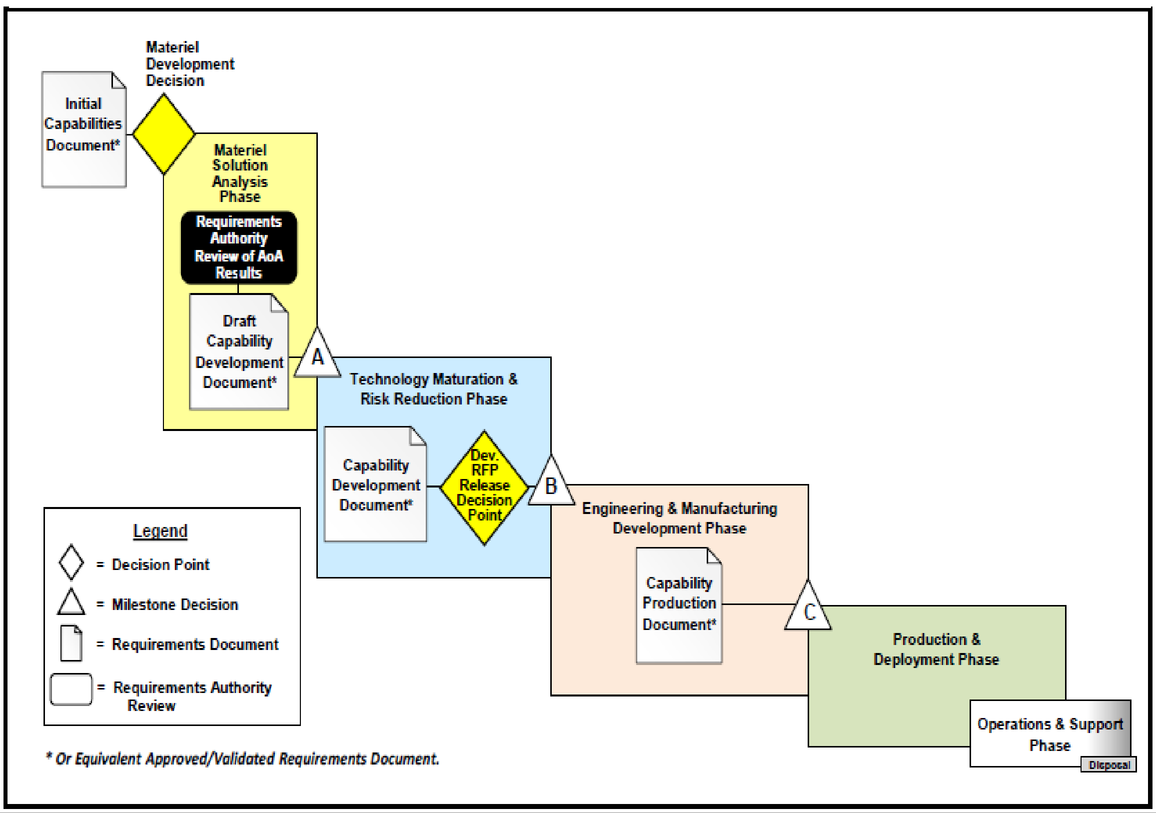


Figure 10. DOD Acquisition life cycle (from USD [AT&L] 2013)

Figure 10 is a graphic depicting the different categories and phases of the Defense Acquisition Management System as defined within the Interim DODI 5000.02 (USD [AT&L] 2013). The Pre-Concept Refinement Phase, shown in

Figure 11 occurs prior to the Materiel Development Decision (MDD). During this phase, the basic principles of a particular technology are observed, reported, and refined. This stage of development is primarily concerned with analytical and experimental activities meant to provide a proof of concept. Once the technology has been proven to be technically feasible and physically achievable, it is generally accepted to be at a Technology Readiness Level (TRL) of 4. TRL definitions and the process for obtaining TRLs are found in Chapter II.1 of this report. The activities that occur during pre-concept refinement are performed by research laboratories such as the DARPA or other DOD service research labs.

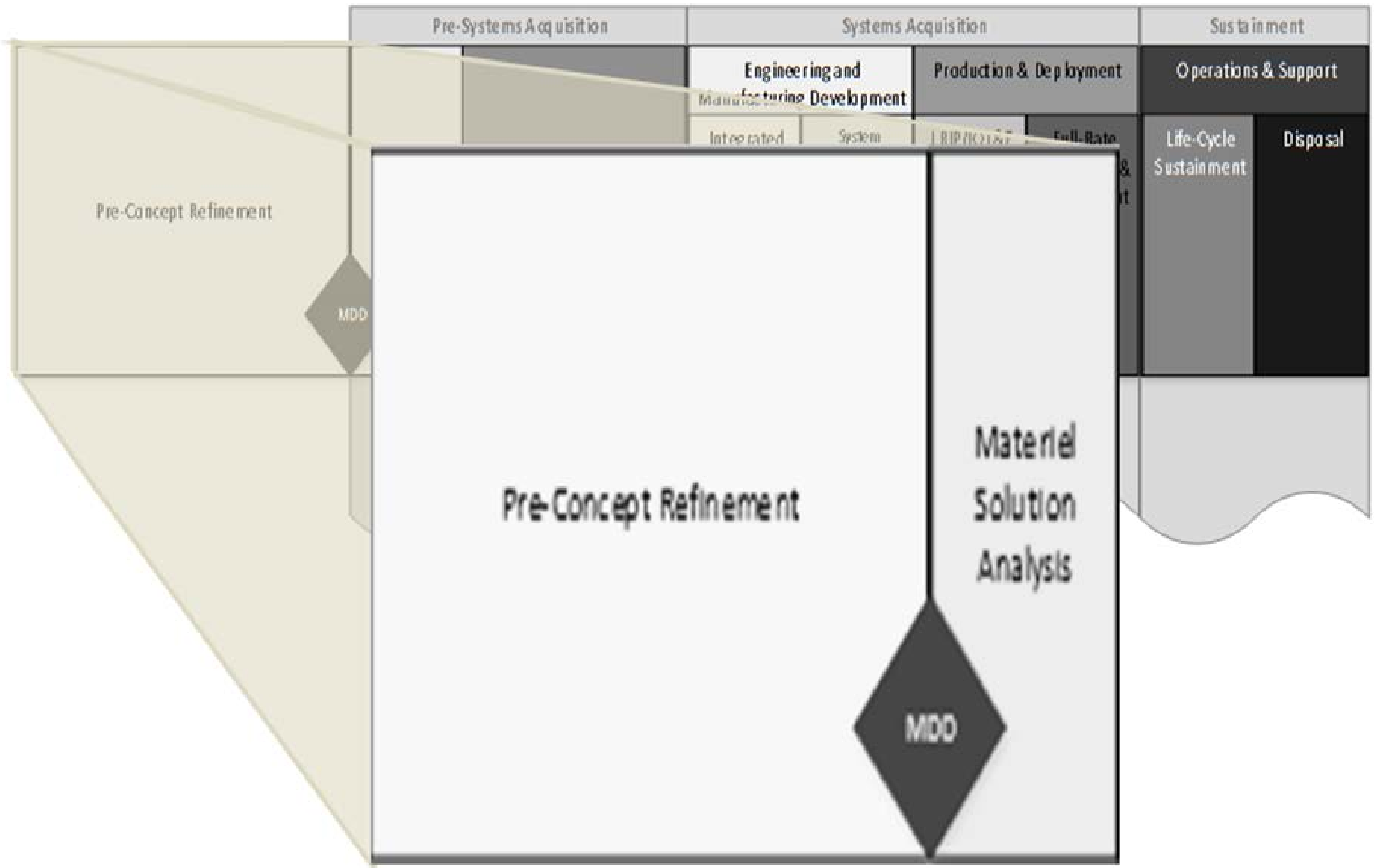


Figure 11. Pre-Concept Refinement (after Interim DOD 5000.2 (USD[AT&L] 2013)

The set of solutions that comprise the TDS will be applicable to the Technology Maturation and Risk Reduction Phase or between MS A and MS B as shown in Figure 13. The TDS will become engaged upon receipt of a service request. The service request is an inclusive term that encompasses user requirements and schedule, program office funding and schedule profiles, and acquisition policies and regulations. The receipt of a service request will trigger a feasibility assessment. The feasibility assessment serves to ensure that all required entrance criteria, for both technical and programmatic aspects, are available and pass the litmus test for executing the requested technological development effort within the given service request constraints. It concludes with the results from a TDS feasibility study. Technological capabilities will not be able to enter into the TDS for maturation until having achieved a TRL of 4. The TRL of the technological capability will be validated and verified with a Technological Readiness Assessment (TRA) during the feasibility assessment.

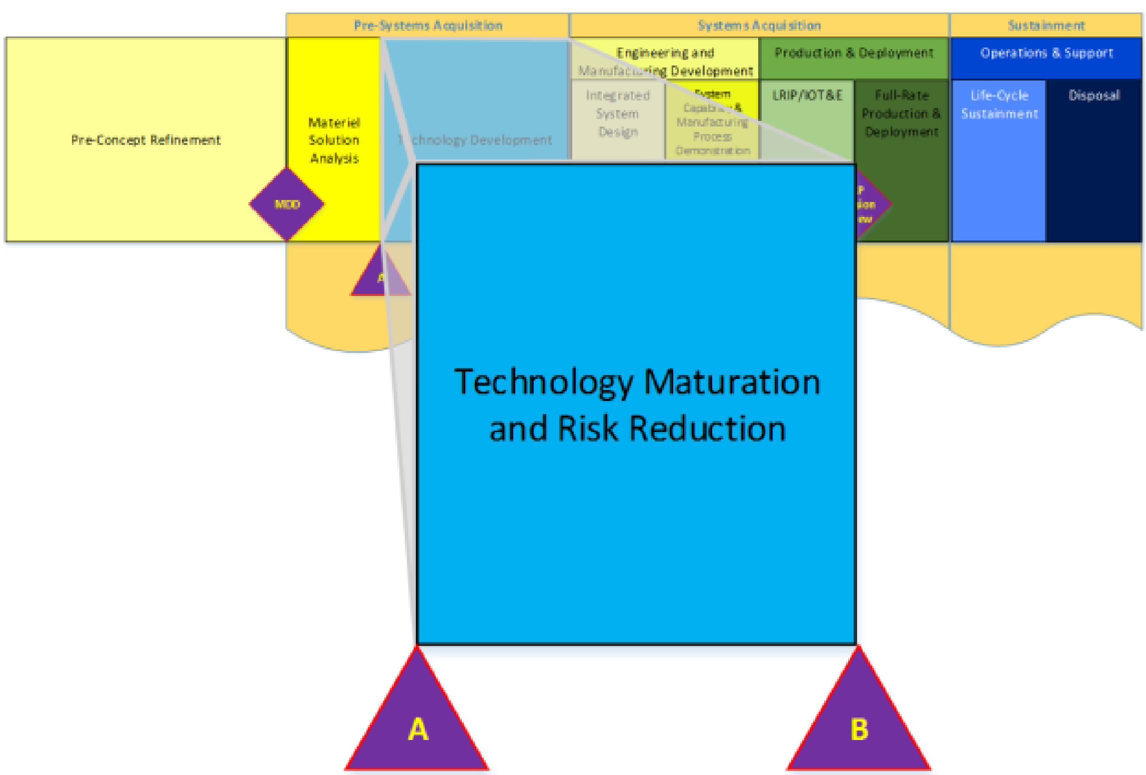


Figure 13. Technology Maturation and Risk Reduction Phase (after Interim DOD 5000.2 (USD [AT&L] 2013))

Once the feasibility of progressing into technological development has been validated, a plan will be developed for maturing the technology. The technology will be matured through detailed planning, design and demonstration of technology or system prototypes, and inter-organizational technology transition agreements between the technical and programmatic entities. Transition of the technology, leaving the TDS and moving into the engineering, manufacturing, and development (EMD) acquisition phase, will occur when it has achieved a TRL 6. By definition, TRL 6 cannot be attained until the technology has been demonstrated in a relevant operational environment (ASD [R&E] 2011). A technology that has matured to a TRL of 6 will be the output of the TDS and serve as input into MS B (Engineering and Manufacturing Development Phase).

A secondary purpose of the TDS is to ensure that all exit criteria have been properly met and documented in order to support a successful transition to the program office that will continue the effort into formal systems acquisition. The TDS is required to work within the boundary and constraints of the DOD 5000, therefore, alignment to the current acquisition construct is key to implementing the improved processes. In order to work within the acquisition guidelines and be aligned with the acquisition milestones, the system will utilize common DOD 5000 language and program documentation in order to allow seamless transition into a program manager's acquisition capability portfolio. The MS A and Preliminary Design Review (PDR) requirements for specific technological capabilities will be addressed and planned as formal reviews with the customer during the normal course of the TDS processes. The MS A requirements, relative to a technological capability, will be captured and reflected as a part of the TDS exit criteria review.

The TDS will be used as a focal point for technology maturation, development planning, and execution. Limiting the TDS to simply advancing technological capabilities without consideration to the programmatic side of acquisition would not yield results conducive to implementation (GAO 2006) Decision makers at many levels of the acquisition construct require information and data to acquire or maintain funding needed for technological capability development; therefore, the TDS will produce a technology transition plan, which will identify relevant program documentation and align technology goals with the program schedule. This will yield a proven technological capability to the program manager enabling the execution of a successful acquisition program.

The TDS will address both system engineering and programmatic concerns, relative to the technological capability, that are required during the Technology Maturation and Risk Reduction Phase. In order to accurately track progress, gate reviews will be implemented and aligned with technology readiness levels (Ellis and Craver 2006). Each gate will be a decision point for the program to move to the next stage of development. These stages or gates will be measured by metrics such as technology risk levels, exit criteria, technology deliverables, and funding.

The TDS, as described, is depicted in Figure 12. The flow progresses from the top- left and flows through each phase, with an exit opportunity occurring at each step. As shown, a technology enters the system and is assessed for feasibility. The technology is first assessed to determine the current TRL. If the appropriate TRL has been achieved, the technology is then assessed from a programmatic standpoint. This assessment ensures the schedule and budgetary constraints are feasible to develop the technology. If the assessments find the project to be infeasible, the technology exits the system and the project ends. If feasible, the project then proceeds to planning technology development. In this phase, cost, risk, and schedule plans are developed for moving the technology to the next step in its maturity. Next, the plan is executed through the creation of prototypes to advance the technology. The technology is then tested and documented. When the technology has reached a TRL of 6, it is ready to transition from the TDS back to the customer and proceed to a MS B review.

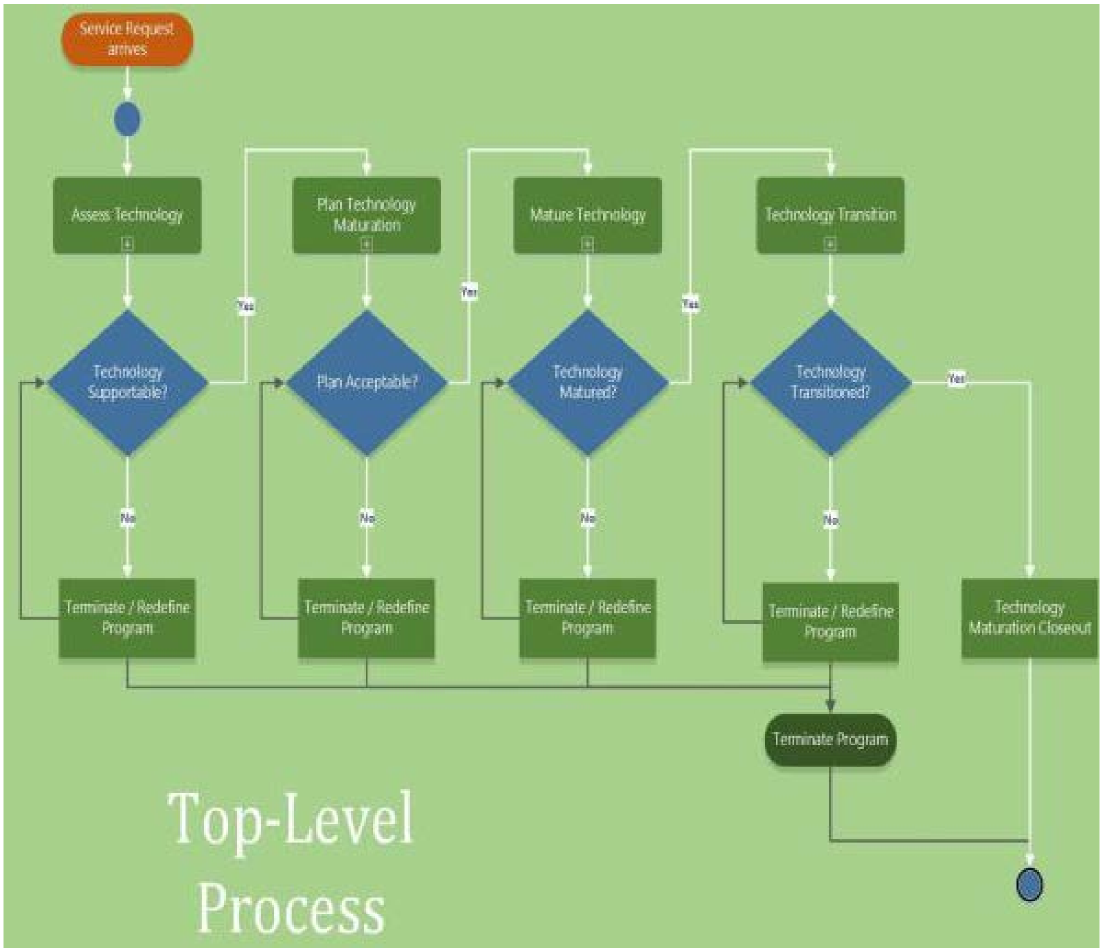


Figure 12. Technology Development System Flow Chart

In summary, the TDS concept will satisfy the functional requirements developed from the stakeholder needs: assess program feasibility, plan for technological development, mature the technology, transition the technology, and closeout the technology maturation process. The TDS will provide the acquisition community with a process model to aid in the successful transition of technology through prototyping. The TDS will be partitioned into phases that utilize gate reviews aligned to technology readiness levels. Detailed activities, entry and exit criteria, and prototyping will be included in the system model. TDS will be used by the program management offices to measure and advance a technology program's development maturity (Ellis and Craver 2006). The TDS will promote early focus on what needs to be completed to effectively transition well defined technology with clearly specified technical risks to the acquisition program customers ensuring that technological capability maturity has occurred and the technology is at an optimum level. As a result, it will provide DOD decision makers with the confidence and knowledge that the capability is ready for integration into the larger system without negatively impacting the overall system development from a cost and schedule standpoint.

1. Technology Readiness Assessments and Levels

The term “TRL” is referenced extensively throughout the operational concept definition. For clarification and terminology, formal definitions for each TRL are presented in Figure 13. TRLs, which range in levels from one to nine, are determined by a TRA. (ASD [R&E] 2011) The TRA is performed by a project office with the assistance of subject matter experts (SMEs). TRAs are required for technologies in any Major Defense Acquisition Program (MDAP). The results of a TRA are provided to the Assistant Secretary of Defense for Research and Engineering (ASD [R&E]) who uses the TRA as one basis for developing input to the Milestone Decision Authority (MDA). (ASD [R&E] 2011)

TRL	Definition	Description	Supporting Information
1	Basic principles observed and reported.	Lowest level of technology readiness. Scientific research begins to be translated into applied research and development (R&D). Examples might include paper studies of a technology's basic properties.	Published research that identifies the principles that underlie this technology. References to who, where, when.
2	Technology concept and/or application formulated.	Invention begins. Once basic principles are observed, practical applications can be invented. Applications are speculative, and there may be no proof or detailed analysis to support the assumptions. Examples are limited to analytic studies.	Publications or other references that out-line the application being considered and that provide analysis to support the concept.
3	Analytical and experimental critical function and/or characteristic proof of concept.	Active R&D is initiated. This includes analytical studies and laboratory studies to physically validate the analytical predictions of separate elements of the technology. Examples include components that are not yet integrated or representative.	Results of laboratory tests performed to measure parameters of interest and com-parison to analytical predictions for critical subsystems. References to who, where, and when these tests and comparisons were performed.
4	Component and/or breadboard validation in a laboratory environment.	Basic technological components are integrated to establish that they will work together. This is relatively "low fidelity" compared with the eventual system. Examples include integration of "ad hoc" hardware in the laboratory.	System concepts that have been considered and results from testing laboratory-scale breadboard(s). References to who did this work and when. Provide an estimate of how breadboard hardware and test results differ from the expected system goals.
5	Component and/or breadboard validation in a relevant environment.	Fidelity of breadboard technology increases significantly. The basic technological components are integrated with reasonably realistic supporting elements so they can be tested in a simulated environment. Examples include "high-fidelity" laboratory integration of components.	Results from testing laboratory breadboard system are integrated with other supporting elements in a simulated operational environment. How does the "relevant environment" differ from the expected operational environment? How do the test results compare with expectations? What problems, if any, were encountered? Was the breadboard system refined to more nearly match the expected system goals?
6	System/subsystem model or prototype demonstration in a relevant environment.	Representative model or prototype system, which is well beyond that of TRL 5, is tested in a relevant environment. Represents a major step up in a technology's demonstrated readiness. Examples include testing a prototype in a high-fidelity laboratory environment or in a simulated operational environment.	Results from laboratory testing of a prototype system that is near the desired configuration in terms of performance, weight, and volume. How did the test environment differ from the operational environment? Who performed the tests? How did the test compare with expectations? What problems, if any, were encountered? What are/were the plans, options, or actions to resolve problems before moving to the next level?
7	System prototype demonstration in an operational environment.	Prototype near or at planned operational system. Represents a major step up from TRL 6 by requiring demonstration of an actual system prototype in an operational environment (e.g., in an aircraft, in a vehicle, or in space).	Results from testing a prototype system in an operational environment. Who performed the tests? How did the test compare with expectations? What problems, if any, were encountered? What are/were the plans, options, or actions to resolve problems before moving to the next level?
8	Actual system completed and qualified through test and demonstration.	Technology has been proven to work in its final form and under expected conditions. In almost all cases, this TRL represents the end of true system development. Examples include developmental test and evaluation (DT&E) of the system in its intended weapon system to determine if it meets design specifications.	Results of testing the system in its final configuration under the expected range of environmental conditions in which it will be expected to operate. Assessment of whether it will meet its operational requirements. What problems, if any, were encountered? What are/were the plans, options, or actions to resolve problems before finalizing the design?
9	Actual system proven through successful mission operations.	Actual application of the technology in its final form and under mission condi-tions, such as those encountered in operational test and evaluation (OT&E). Examples include using the system under operational mission conditions.	OT&E reports.

Figure 13. TRL Definitions (from ASD [R&E] 2011)

Two TRLs, which will be referenced throughout this report, include TRL 4 and TRL 6. As detailed in Figure 13, TRL 4 represents the basic components, which have been integrated and validated in a laboratory environment. To achieve a TRL of 6, a prototype will be required to complete a demonstration in a mission representative environment.

F. SUMMARY

In Chapter II the team provided an evaluation of the stakeholder analysis and discussed much of extensive research that was accomplished specific to DOD Acquisition and prototyping. In order to understand the stakeholder needs, Team BlackberryPI analyzed the problem statement developed from the literature review and the importance of the need and the problem. The team conducted an analysis of the basic stakeholder needs in order to refine those needs and develop system-level requirements. The functional requirements developed expressed “what” functions the system must do in order to meet the stakeholder needs. The Operational Concept and technical approach was also developed and iterated until the team had achieved an operational concept that would meet required system-level requirements. Developing the functional requirements and conceptual system operation were key steps toward refining the system requirements, developing the system functional hierarchy, and establishing the evaluation measures of the system. The functional hierarchy, evaluation measures and value hierarchy will be described and discussed in Chapter III.

III. VALUE SYSTEM DESIGN AND FUNCTIONAL ANALYSIS

The systems engineering process, to this point, has been focused on refining the problem, establishing the top-level system requirements, and developing the overall system concepts. All of the aforementioned activities were conducted in order to perform the functional analysis and value system design. The functional analysis was performed in order to identify and decompose the system functions, as well as to develop the functional hierarchy. The identification and hierarchical order of the system functions were important to understanding the full functional implementation of the system concept. The Integrated Definition 0 (IDEF0) function modeling method was used to model the decisions, actions, and activities of the system in order to communicate the functional perspective of the TDS (Colquhoun, Baines and Crossley 1993). The IDEF0 model was created as part of the system development in order to describe the functions to be performed by the TDS and what processes, resources, and data inputs are needed to perform those functions. Identifying the mechanisms and system relationships aided in the development of the executable model in the next phase of the SE process. In addition, the functions and functional mechanisms were decomposed to build the full list of system requirements. Next, a value system and its associated metrics were developed to evaluate the TDS. The value system design identified objectives for each of the system functions that are directly related to the customer "wants" discussed in the stakeholder analysis. After defining the objectives, the team developed measures to evaluate the degree to which the systems have met those objectives. These evaluation measures provided the key metrics for validating the TDS concept during the simulation modeling phase of the SE process.

A. FUNCTIONAL ANALYSIS

The fundamental purpose of the functional analysis was to decompose the top-level system function into its supporting functions. These supporting functions were derived from the top-level system functional requirements and decomposed to the lowest level necessary to explain and implement the primary executable activities of the system. (Blanchard and Fabrycky 2011)

During functional decomposition, analysis was performed to identify and describe the functional elements and associated interactions of the system. This analysis also aided in identifying the inputs, controls, outputs, and mechanisms (ICOMs) of each functional element. The team utilized a MBSE approach for this project. MBSE is fundamentally a thought process, which utilizes models to allow a systems engineering team to be effective and consistent from the very beginning of a project. This process, as described by Long and Scott and utilized by the team, is a layered problem-solving process that begins at the highest and most general layer of the system and looks at the problem statement, requirements, architecture, validation and verification at that level before moving to the next more “granular level” and utilizes a set of tools to develop models for each of those layers and activities. (Long and Scott 2011)

Functional analysis was an iterative and recursive process, which allowed the team to arrive at a complete hierarchical breakdown of the functional activities of the system. While functional modeling and analysis did not address how the functions would be performed, it did identify and relate the functions that the TDS must perform to meet stakeholder needs. This architecture was also utilized to indentify and understand the relationships between each of the system functions. (Blanchard and Fabrycky 2011)

The team utilized the Innoslate cloud-based SE software designed by SPEC Innovations to perform the functional analysis of the TDS system. The Innoslate SE software tool uses the MBSE approach for development, allocation and management of system functions and requirements and provides SE teams the ability to create diagrams depicting the functional elements and relationships of the respective system. The BlackberryPi team selected the Innoslate tool because it provided an intuitive approach for functional analysis and management, system concept diagram creation and system simulation combined with team and advisor collaboration capability over the Internet. (SPEC 2012)

1. Top-level System Functions

The first step of the functional analysis was to develop the top-level functions. These functions were derived from the needs analysis and research. The team began functional decomposition by asking the question "What does the system have to do?" Extensive research and analysis was conducted to identify the top-level functions that would satisfy the operational concept. In the operational concept TDS is a black box system transforming inputs into outputs. The team reviewed the activities that the TDS had to perform to produce the required output and identified its simple systems functions. These functions were then aggregated as appropriate to form top-level functions (Buede 2009). This research led to an understanding that an accurate assessment of the initial maturity of the technology is very important to successful technology development (GAO 2012). Further, an accurate understanding of the programmatic expectations from the stakeholders including cost, schedule, and performance factors was determined to be a necessary function of the system (GAO 2012; GAO 2006). Research revealed that a technology development effort should begin with the desired end state in mind. Identifying the end goal of the development effort at the beginning serves to control the effort. The agreement by primary stakeholders as to the desired outcome of the technology development effort allows control of the supporting activities to only those required to mature the technology. (Borowski 2012). Finally, the research showed that the ability for the matured technology to be transitioned successfully has been a barrier for DOD technology development success (Borowski 2012; GAO 2006). This research showed the critical functions for TDS based on the operational concept and was aggregated to form a top-level system function (Buede 2009) of Perform DOD Prototyping. This top-level function was decomposed into six system functions as follows:

- Assess Feasibility
- Produce Technology Development Plan
- Mature Technology
- Redefine/Terminate Program
- Transition Technology

- Technology Maturation Closeout

Identification of these six primary system functions provided the ability to construct the functional architecture and continue to decompose the system to further layers of granularity. These processes were developed iteratively and recursively and each iteration increased the specificity, removed ambiguity, and resolved unknowns about the systems functionality (Long and Scott 2011; Buede 2009; Blanchard and Fabrycky 2011). The team continued performing this iterative decomposition until it reached a level where further decomposition would require assumptions that were no longer supported by the operational concept (Buede 2009). Functional interactions within the hierarchy will be presented first, followed by the functional architecture.

2. Functional Decomposition and Hierarchy

The second-level functions, shown in Figure 14, represent the functional applications of the system.

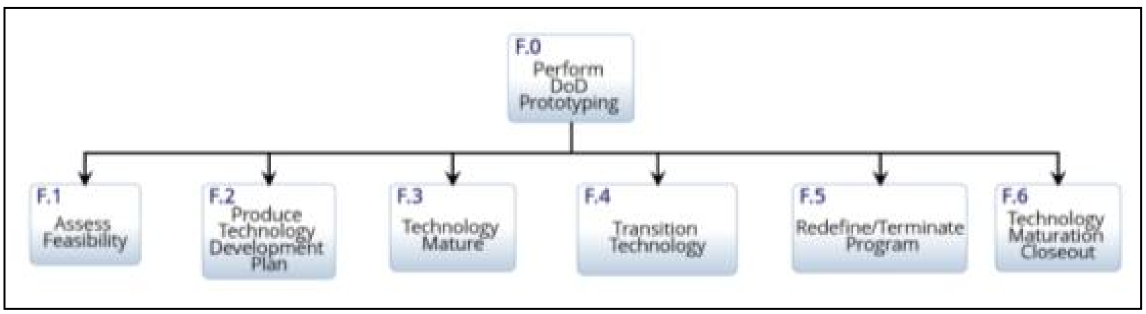


Figure 14. Functional Hierarchy

The functional decomposition was synthesized into a hierarchical format to provide a foundation for developing an understanding of the necessary functions of the system. This allowed the presentation of the constituent parts of the system and provided insight into the identity of the functions. The interrelated behavior of these functions will be discussed in further detail in the Functional Architecture section.

Assess Feasibility was further decomposed as shown in Figure 15. This function served to assess the overall feasibility of the technology being requested for maturation. There were three supporting functions decomposed from Assess Feasibility:

- Technology Readiness Assessment: This sub-function served to ensure that the TRL of the technology entering the system is accurately assessed and able to be matured within the constraints of the TMRR phase of acquisition.
- Assess Technical Feasibility: This sub-function required an assessment of the technical feasibility of maturing the technology entering the system.
- Assess Programmatic Feasibility: This sub-function required an assessment to be performed to determine if the programmatic constraints of the customer are sufficient to allow for the maturation of the technology within those constraints.

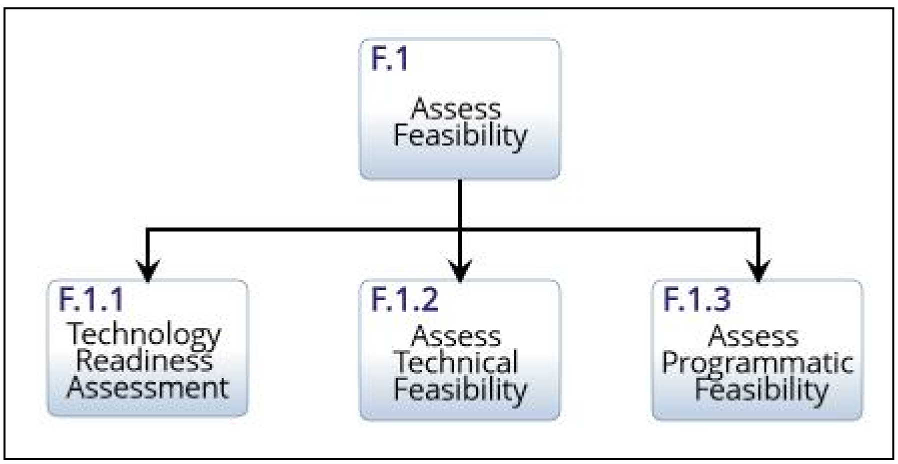


Figure 15. Assess Feasibility Functional Decomposition

Produce Technology Development Plan is decomposed as shown in Figure 16. This function produces a plan that will document how the technology will be developed and matured. The desired output of this function is a Transition Development Agreement, which is signed by the technology developer and the customer. There were four supporting functions decomposed from Produce Technology Development Plan:

- Determine Maturation Risks for the Next Phase: This sub-function requires an assessment of the current risks that have been defined for the technological development and the identification of anticipated/projected risks for the technological maturation.

- Determine Maturation Costs for Next Phase: This sub-function is needed to produce a cost estimate to mature the technology from its current TRL to the next.

- Determine Maturation Schedule for Next Phase: This sub-function produces a schedule estimate for maturing the technology from its current TRL to the next.

- Finalize Plan for Agreement: This sub-function produces the output of the finalized plan for maturing the technology from its current TRL to the next that is acceptable to the developer and the customer.

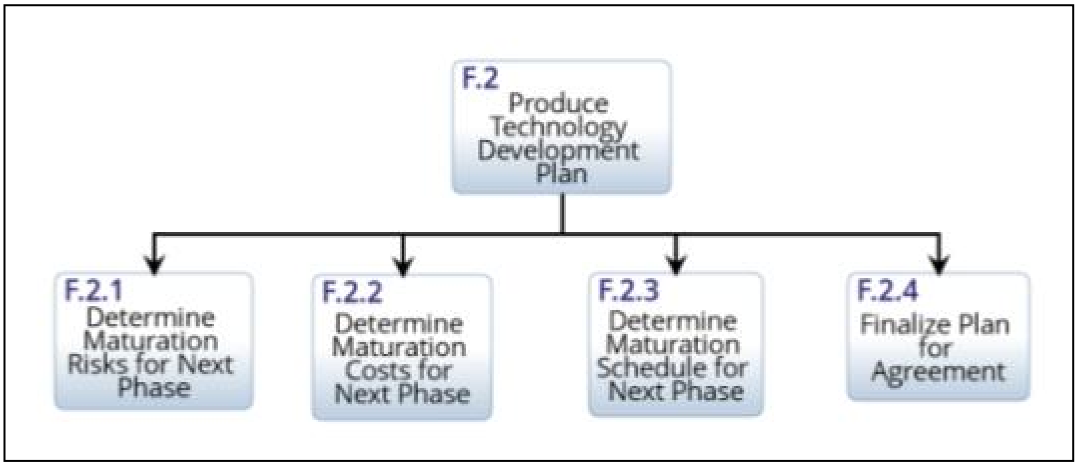


Figure 16. Produce Technology Development Plan Functional Decomposition

Mature Technology was decomposed as shown in Figure 17. This function is the focal point of the system where the technology maturation process takes place. The Technology Development Plan is used to define how the technology maturation activities will be executed during this function. The technology to be matured is expected to enter at a TRL 4 and be matured to a TRL 5, and subsequently, to a TRL 6. There are four supporting functions for Mature Technology:

- Design Prototypes: This sub-function encompasses the processes and activities required to design a prototype.
- Build Prototypes: This sub-function is responsible for building the prototype in accordance with the design generated by the preceding Design Prototypes function.
- Demonstrate Prototype in Simulated Environment: This sub-function serves to perform a demonstration of the prototype in a simulated environment.
- Demonstrate Prototype in Operational Environment: This sub-function serves to perform a demonstration of the prototype in an operational environment.

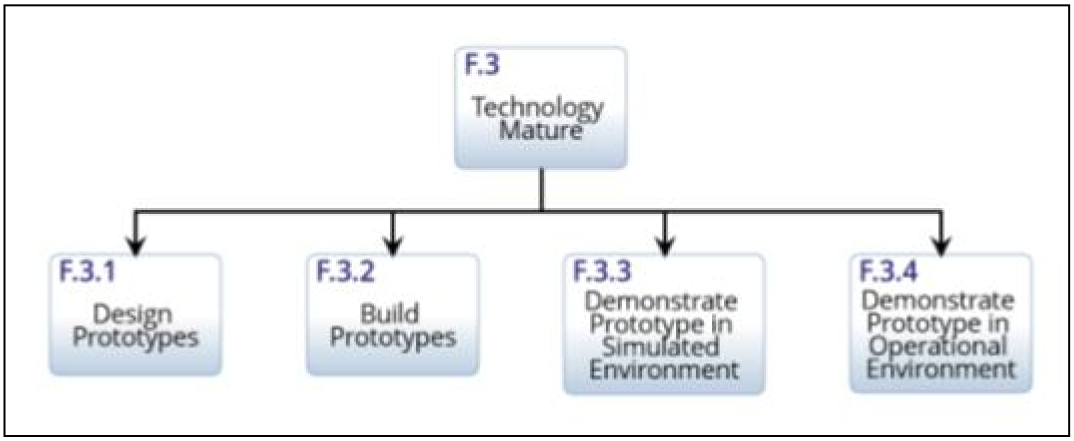


Figure 17. Mature Technology Functional Decomposition

Design Prototypes was decomposed into nine sub-functions as shown in Figure 18.

- Define System Boundary: This sub-function serves to define the system boundary of the technology under development.
- Derive System Threads: This sub-function serves to derive and define the system threads for the technology under development.

- Derive Component Hierarchy: This sub-function serves to derive and define the component hierarchy for the technology under development.
- Allocate Behavior to Components: This sub-function serves to allocate the technological behaviors to the components that were defined under the previous function.
- Perform Modeling and Simulation: This sub-function serves to perform and execute modeling and simulations on the different designs generated by the preceding functions.
- Perform Effectiveness and Feasibility Analysis: This sub-function serves to evaluate the results produced by the modeling and simulation performed by the previous function.
- Select Design: This sub-function serves to select the best design for the capability/technology based on the analysis and results of the modeling and simulation data.
- Define Resources, Error Detection, and Recovery: This sub-function serves to define the proper amount of resources, level of error detection, and recovery required for the development of a prototype.
- Generate Documentation and Specifications: This sub-function serves to ensure that the design documentation and specifications have been generated and are in the proper format.

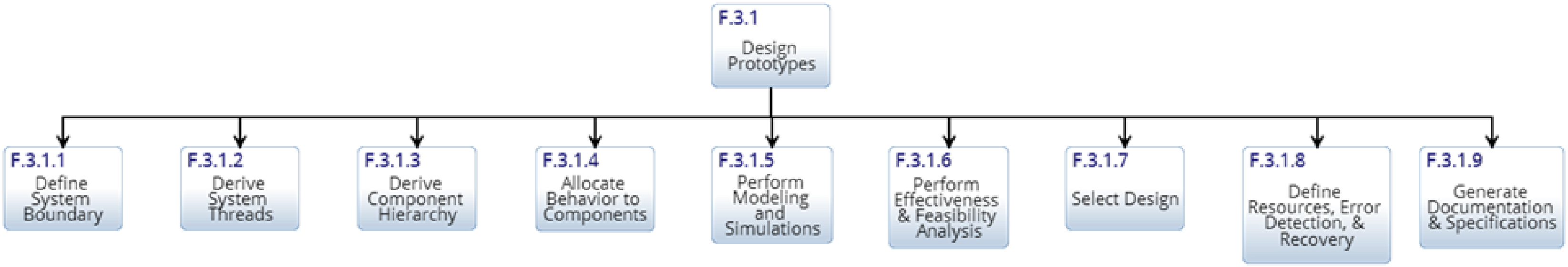


Figure 18. Design Prototypes Functional Decomposition

Build Prototypes is decomposed into four sub-functions as shown in Figure 19.

- Build Prototype Hardware: This sub-function builds/produces a hardware prototype based on the design selected.
- Build Prototype Software: This sub-function builds/produces a software prototype based on the design selected.
- Integrate the Prototype Components: This sub-function integrates the hardware and software prototype components produced by the previous functions.
- Perform Component Integration Testing: This sub-function verifies and validates the integration of a prototype using the required testing methods.

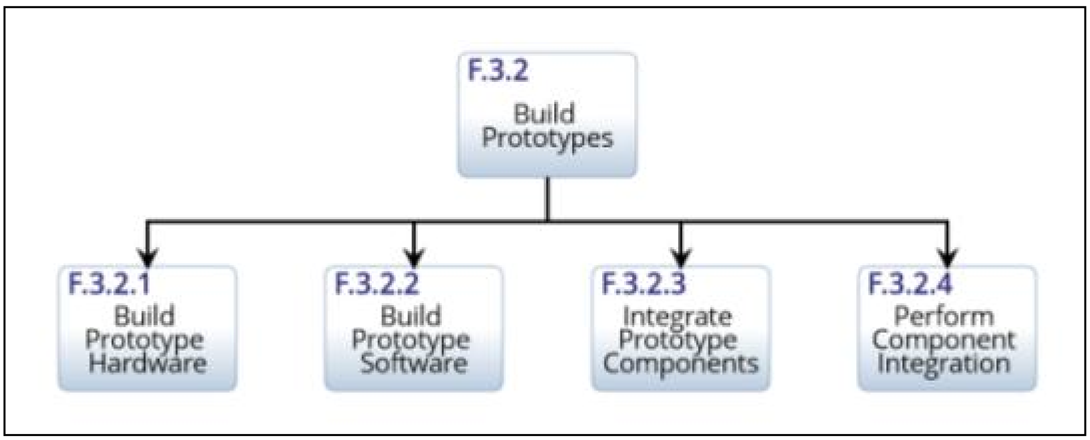


Figure 19. Build Prototypes Functional Decomposition

Demonstrate Prototype in Simulated Environment was decomposed into three sub-functions as shown in Figure 20.

- Model Simulated Environment: This sub-function serves to build/produce an approved simulated environment for prototype testing.
- Run Prototype in Simulated Environment: This sub-function is responsible for the preparation and execution of a prototype within a simulated environment. Pertinent data is also collected during the simulation.

- Evaluate Results: This sub-function serves to evaluate the results and data that were produced by the prototype under test within the simulated environment.

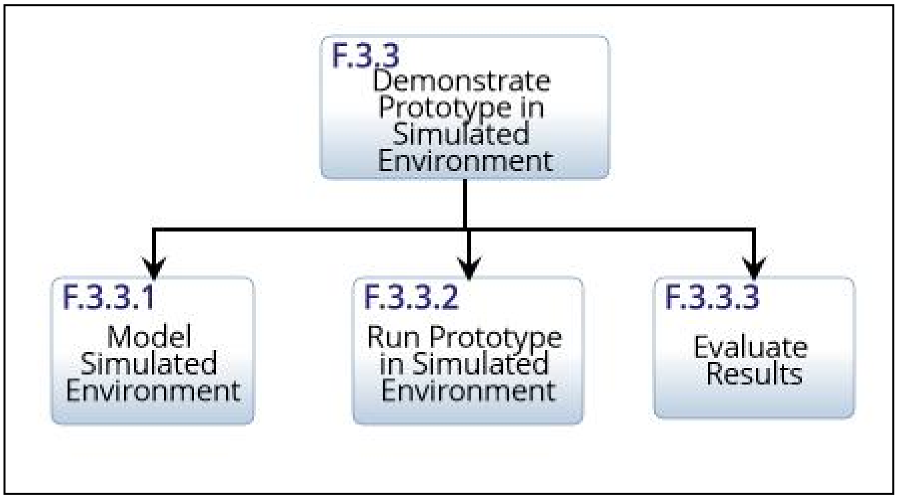


Figure 20. Demonstrate Prototype in Simulated Environment Functional Decomposition

Demonstrate Prototype in Operational Environment was decomposed by three sub-functions as shown in Figure 21.

- Validate Operational Environment: This sub-function serves to validate the operational environment in which the technology under development will be tested.

- Demonstrate in Operational Environment: This sub-function is responsible for the preparation and execution of a prototype within an operational environment. Pertinent data is also collected during the demonstration.

- Evaluate Results: This sub-function serves to evaluate the results and data that were produced by the prototype under test within the operational environment.

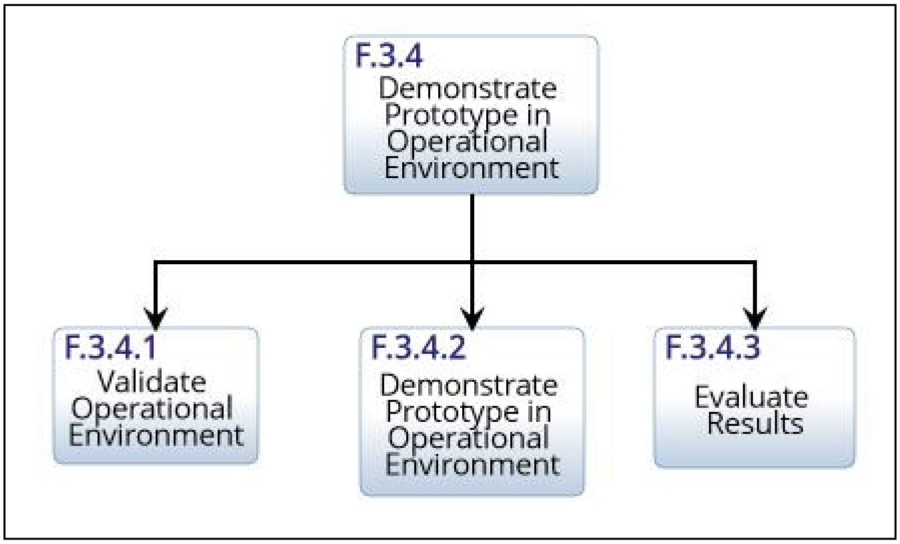


Figure 21. Demonstrate Prototype in Operational Environment Functional Decomposition

This concludes the fourth level decomposition of the Mature Technology function. At this stage there was not further value added by decomposing the functions further. The decomposition to this point adequately describes the functionality and the production of the system outputs as described in the concept of operations (Buede 2009). The following paragraphs will resume discussion of the remaining second level functions of the system.

Transition Technology was decomposed into three sub-functions as shown in Figure 22. This function determines whether the technology under development is ready to transition from the DOD Prototyping System to the next phase of acquisition development.

- Finalize Technology Transition: This sub-function serves to finalize the artifacts produced during technology development in order to support the Transition TRA.
- Perform Technology Readiness Assessment: This sub-function serves to assess and ensure that the out-going technology has matured to a TRL of 6.

- Transition Technology Artifacts: This sub-function serves to assemble associated system artifacts and illustrate the progress of the technology to TRL 6.

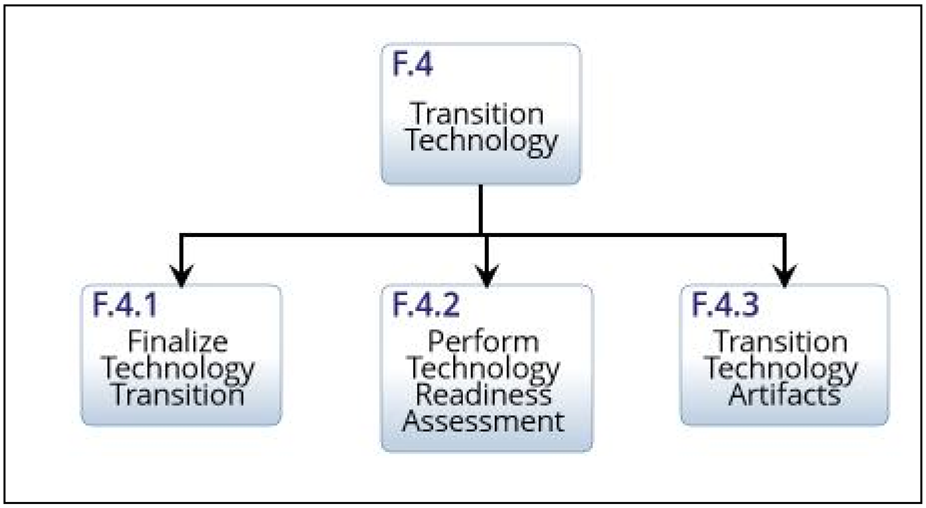


Figure 22. Transition Technology Functional Decomposition

Redefine/Terminate Program was decomposed by three sub-functions as shown in Figure 23. This function serves to determine whether the plan for the technology under development should be redefined or if the program should be terminated based on available data.

- Program Determination: This sub-function serves as an assessment of the overall status of the program to determine if the program needs to be re-baselined or terminated.
- Redefine Program Plan: This sub-function serves to provide notification that the program will be redefined.
- Capture Issue Metrics: This sub-function serves to determine and collect issue metrics associated with faults or failures within the system. This information would be utilized to support process improvements.

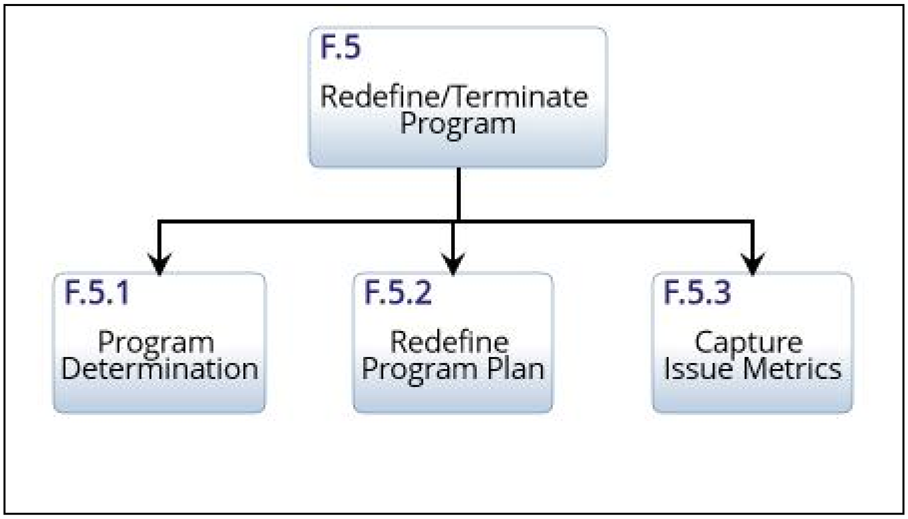


Figure 23. Redefine/Terminate Program Functional Decomposition

Technology Maturation Closeout was determined to be a standalone function as shown in Figure 24. This function serves as the final closeout operation of the DOD Prototyping System. Regardless of whether the technology maturation has been successful, all technology development will go through an official closeout process. This process was meant to capture all of the associated artifacts of the technology development activities and passed to the customer as a service response.

Figure 24. Technology Maturation Closeout Functional Decomposition

Functional Decomposition Summary: The functional decomposition was completed to describe the system to the level described in the concept of operations. A MBSE approach was used to create the decomposition as described by Scott and Zane using the Innoslate tool. This decomposition was then used as a tool to show in a hierarchical view how the functions decompose. The team utilized this decomposition to create the system architecture as described by Buede and as shown in the following section (Buede 2009).

3. Functional Architecture

As each level of the functional hierarchy was established, the team assessed the relationships among the decomposed functions. Evaluating the relationships that occur between functions of the system provide traceability and understanding of the key elements required to accomplish the individual system functions while also providing a holistic view of how the system accomplishes the top-level function. The team used the IDEF0 method to develop the model of the functions. The IDEF0 model was used to illustrate the implementation and interaction of system functions and provide an identity for the elements necessary for operation of each function. These elements were identified as the inputs, outputs, controls, constraints and mechanisms of each function (Buede 2009).

a. Context Architecture

The functional architecture as a logical model that captured the transformation of inputs into outputs began by creating an IDEF0, as depicted in Figure 25. IDEF0 modeling diagrams were developed when the United States Air Force commissioned the developers to develop a function-modeling tool for visually displaying data to analyze and communicate the functional perspective of a system. The models are used for decision-making, identifying functions, function performers and function actors. (Knowledge Based Systems 1993). This functional architecture model resulted in a system context to identify the boundaries of the system, sources for the inputs, and destinations for the outputs. The information from the needs analysis was utilized to develop this diagram and provided the starting point for the functional architecture.

There were three categories of external functions identified: Perform Government Entities, Perform End User Activities, and Perform Customer Activities. These activities are sources for external inputs and controls, as well as, the destinations for the outputs of the TDS. The system is triggered by the control identified as a service request from the customer. The customer is the person or group who originally requested the system or process, defines the overall objectives, provides basic requirements, and usually coordinates funding for the project. (Pressman 2010) This service request is a combination of inputs, including funding profile, schedule, requirements, etc., and provides the initiating trigger for the technology system functions. This control is intentionally broad in scope to allow tailoring based on the variable inputs that will be generated within the external system functions.

The TDS framework was built to account for variability based on the scope of the technology being developed. In order to produce this service request, the customer will require input from the end users of the technology for schedule and requirements as well as other government entities to provide funding levels. The end user is the warfighter or person who uses the system or process. (Pressman 2010) The government also provides overarching regulations and policies serving as controls on the development processes of the technology. Throughout the development process there will be interaction with the end user to receive clarification of requirements submitted as part of the Service Request. A request for Requirements Clarification will be one output of the system. The primary output is the Service Responses. Service Responses will serve as the bulk of the interactions between the system and the customer, including plans, reports, and status updates. The Service Responses are a collaborative output, agreed to by all parties during the TDS Activities, and are detailed in the following sections.

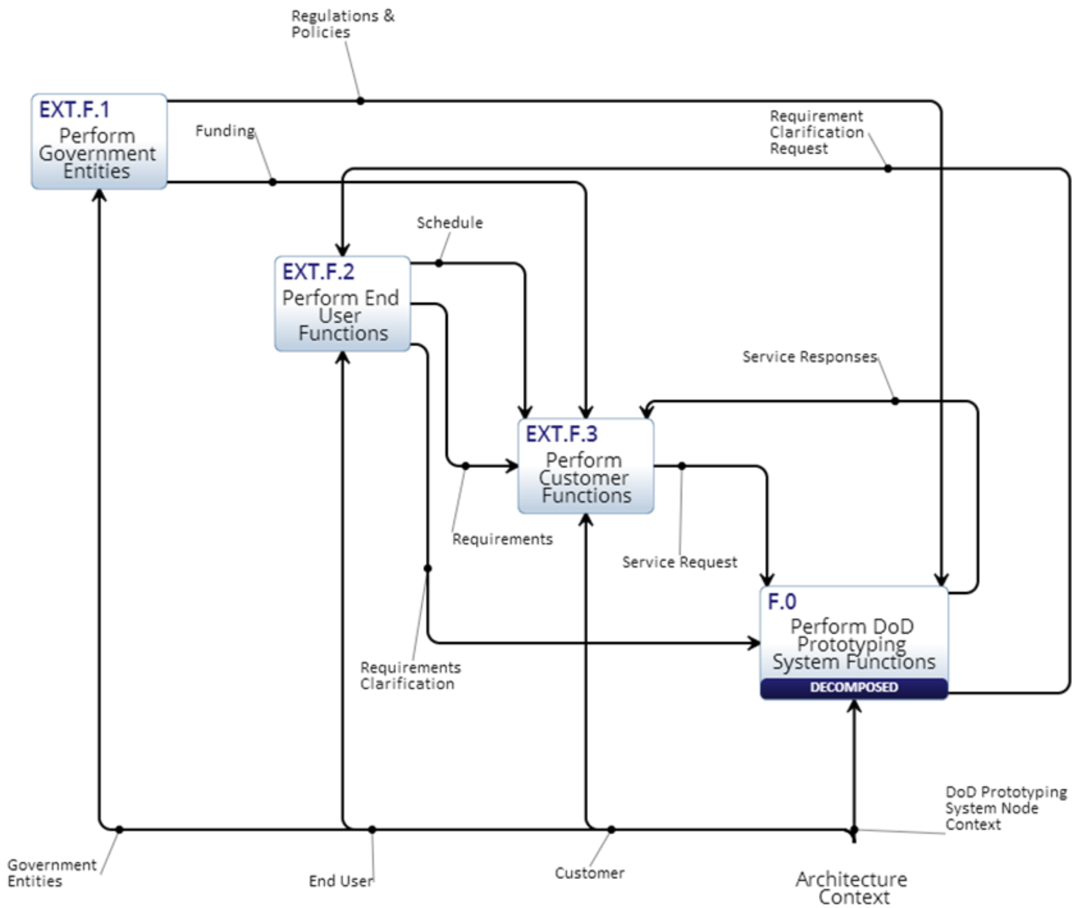


Figure 25. IDEF0 for TDS System Context

b. Top-Level Architecture

As stated earlier, functional decomposition of the system identified six top-level functions that form the functional architecture shown in Figure 26. This figure is meant to provide the viewer with the relationships and traceability of lower level functional inputs, controls, outputs, and mechanisms for each function internal to the system.

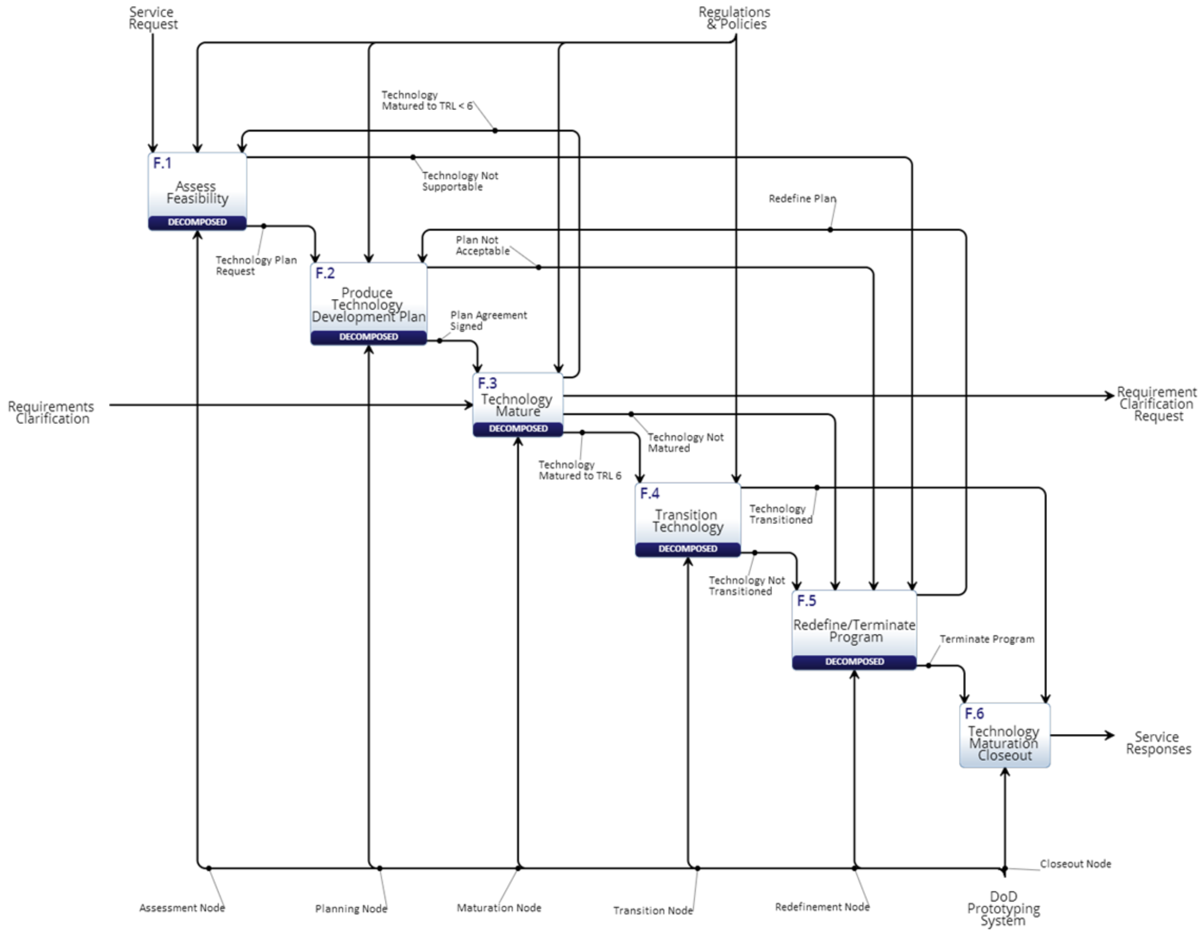


Figure 26. Top-Level IDEF0

As shown in the IDEF0, Assess Feasibility has two controls identified as Service Request and Regulations and Policies. Regulations and Policies describe the guidelines that govern an acquisition development effort; therefore it serves as a control to each function of the system.

Assess Feasibility is meant to denote assessing the technological and programmatic feasibility of a specific technology that enters the system. Two outputs are possible for this function: Technology Not Supportable or Technology Supportable. The mechanism for the Assess Feasibility function is the Assessment Node. The purpose of this function is to determine if maturing the technology is feasible given the programmatic constraints and current level of technological maturity from the Service Request. This is an initial screening to ensure that resources are not expended on technologies that are clearly not ready and to identify, in advance, a program that is not executable. If the technological maturity or programmatic constraints make technology maturation infeasible, the output will be Technology Not Supportable and will serve as a control into Redefine/Terminate Program. If the program is deemed feasible from a technological and programmatic standpoint, the system will provide a Technology Plan Request as an output and a control to the Produce Technology Development Plan function.

Produce Technology Development Plan had two possible outputs: Plan Agreement Signed or Plan Not Acceptable. The purpose of this function was to create a plan for the program to mature the technology including coordination with stakeholders to ensure agreement with the necessary milestones to achieve the TRL 6 and subsequent completion of the maturation process.

The Technology Development Plan will include entry and exit criteria for each stage of maturation, documented gate review metrics, as well as reporting criteria for cost, risk, and schedule. The Mechanism for this function is the Planning Node. An output of Plan Agreement Signed will serve as a control for the Mature Technology Function. Conversely, an output of Plan Not Acceptable will serve as a control for Redefine/Terminate Program.

The Plan Agreement Signed control will trigger the Mature Technology function to initiate. There are three possible outputs for this function: Technology Matured to TRL 6, Technology Not Matured, and Requirements Clarification Request. The Requirements Clarification Request is a system level output that serves as a control to trigger Perform End User Activities. Based on a request for requirements clarification, Perform End User Activities will then reciprocate with requirements clarification as an input to the system and to Mature Technology.

Technology Matured to TRL 6 is a control to trigger the Transition Technology function. This is the ultimate goal of the TDS – to mature a technology to TRL 6 and transition that technology into formal system development. The Technology Not Matured output will serve as a control to trigger the Redefine/Terminate Program for program re-assessment.

Transition Technology is triggered by the Technology Matured to TRL 6 control. There are two possible outputs for the Transition Technology Function: Technology Transitioned, and Technology Not Transitioned. The output titled Technology Transitioned will serve as a control to trigger the Technology Maturation Closeout function and begin the actions necessary for transitioning the technology to the Customer. The output titled Technology Not Transitioned will serve as the control for Redefine/Terminate Program and trigger the activities for program re-assessment.

The Redefine/Terminate Program function can be controlled, or triggered, four different ways: Technology Not Supportable, Plan Not Acceptable, Technology Not Matured, and Technology Not Transitioned. All of these controls serve as a trigger for some type of action within the function. The outputs of Redefine/Terminate Program include Terminate Program that will trigger activity within Technology Maturation Closeout, or Redefine Plan, which iterates back to Produce Technology Development Plan.

Technology Maturation Closeout is controlled, or triggered, by either Terminate Program or Technology Transitioned and will produce a system level output in the form of a Service Response back to Perform Customer Activities.

c. Second-Level Architecture

The next level of the architecture will show the decomposition of each top-level function. These model views illustrate the relationships of the functions based on their inputs, outputs, controls, and mechanisms. The decomposition into the lower level functions is meant to ensure that functional boundaries are assessed and the requirements for the system will address each of the functions to ensure the system will perform adequately.

(1) Assess Feasibility. The Assess Feasibility Function, shown in Figure 27, is decomposed by three sub-functions: Technology Readiness Assessment, Assess Technical Feasibility, and Assess Programmatic Feasibility. The overall purpose of these functions is to examine technology maturation program requests for attributes indicative of success.

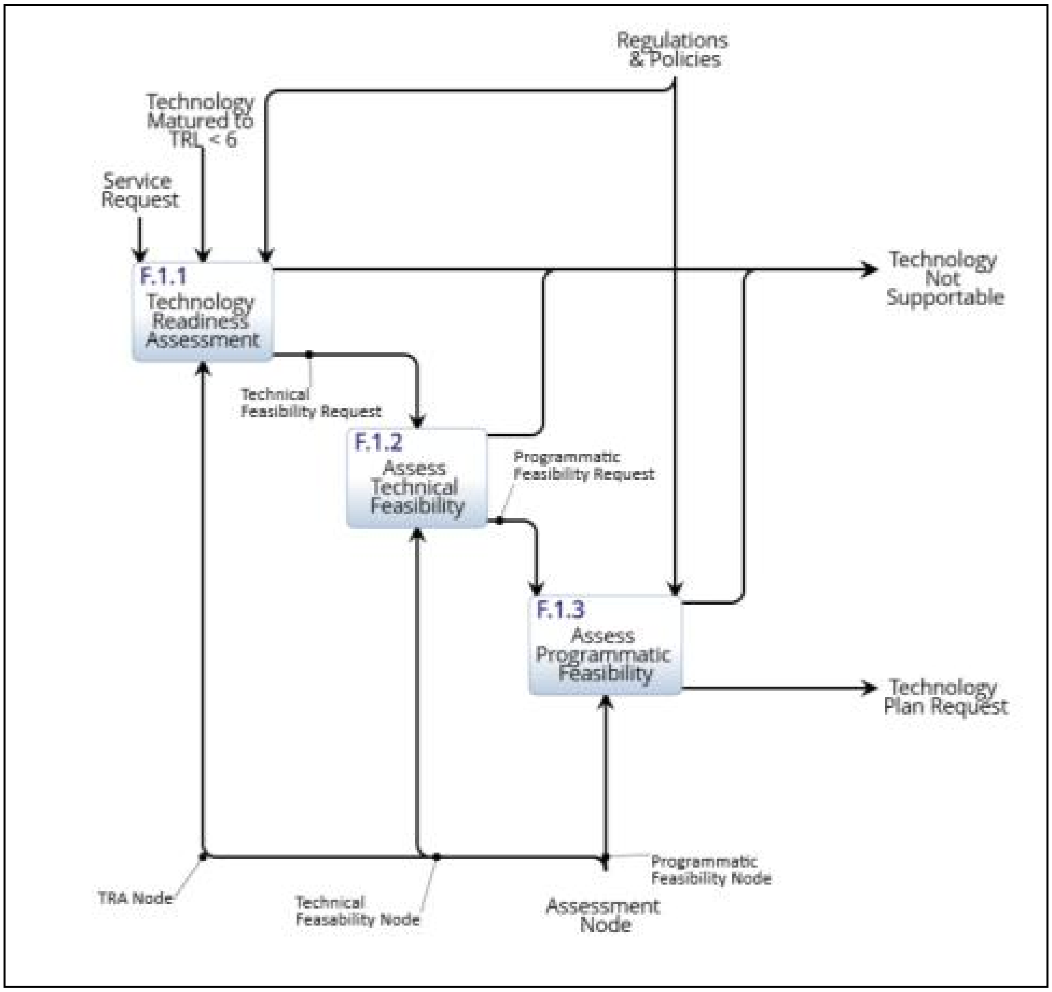


Figure 27. Assess Feasibility IDEF0

There were two primary controls determined to initiate activity within this function: "Service Request" and "Technology Matured to TRL<6." The "Service Request" control is received from the customer and triggers initiation of the Technology Readiness Assessment sub-function. This activity combines the necessary steps to determine if the TRL of the incoming technology meets the entrance criteria required by the system.

Two primary outputs of the Technology Readiness Assessment sub-function were determined to be: "Technology Not Supportable" and "Technical Feasibility Request." "Technology Not Supportable" indicates that a justifiable determination has been made that the technology does not possess the required attributes indicative of successful maturation within the context of the system. This output will serve as a control to initiate activities within the Technology Maturation Closeout function. “Technical Feasibility Request” indicates that the technology entering the system has achieved TRL 4. This output serves as a control to trigger initiation of the Assess Technical Feasibility sub-function.

It was determined that the Assess Technical Feasibility sub-function performs an assessment of the technical feasibility of maturing the technology being introduced into the system. The intended goal of the maturation process and the current status of the technology provide the gap to be analyzed. This function serves to determine whether the required resources are available to attain the intended technology end state. There are two possible outputs for this sub-function: “Technology Not Supportable” and “Programmatic Feasibility Request.” As discussed in the previous paragraph, “Technology Not Supportable” is sent as a control to initiate activities within the Technology Maturation Closeout function. If the technical feasibility is validated, “Programmatic Feasibility Request” is sent as a control to initiate the activities within the Assess Programmatic Feasibility sub-function.

Upon receipt of the programmatic feasibility request, Assess Programmatic Feasibility performs an assessment to determine if programmatic constraints, to include cost and schedule, are feasible for maturing the technology to TRL 6. This assessment analyzes the ability of the system to complete the project successfully. There are two possible outputs for this sub-function: “Technology Not Supportable” and “Technology Plan Request.” “Technology Not Supportable” is sent as a control to trigger activities internal to the Technology Maturation Closeout function. “Technology Plan Request” serves as the control to trigger activities internal to the Produce Technology Development Plan function.

(2) Produce Technology Development Plan. The Produce Technology Development Plan function, shown in Figure 28, was decomposed into four sub-functions: Determine Maturation Risks for Next Phase, Determine Maturation Costs for Next Phase, Determine Maturation Schedule for Next Phase, and Finalize Plan for

Agreement. This function serves to produce a planned approach for developing the technology.

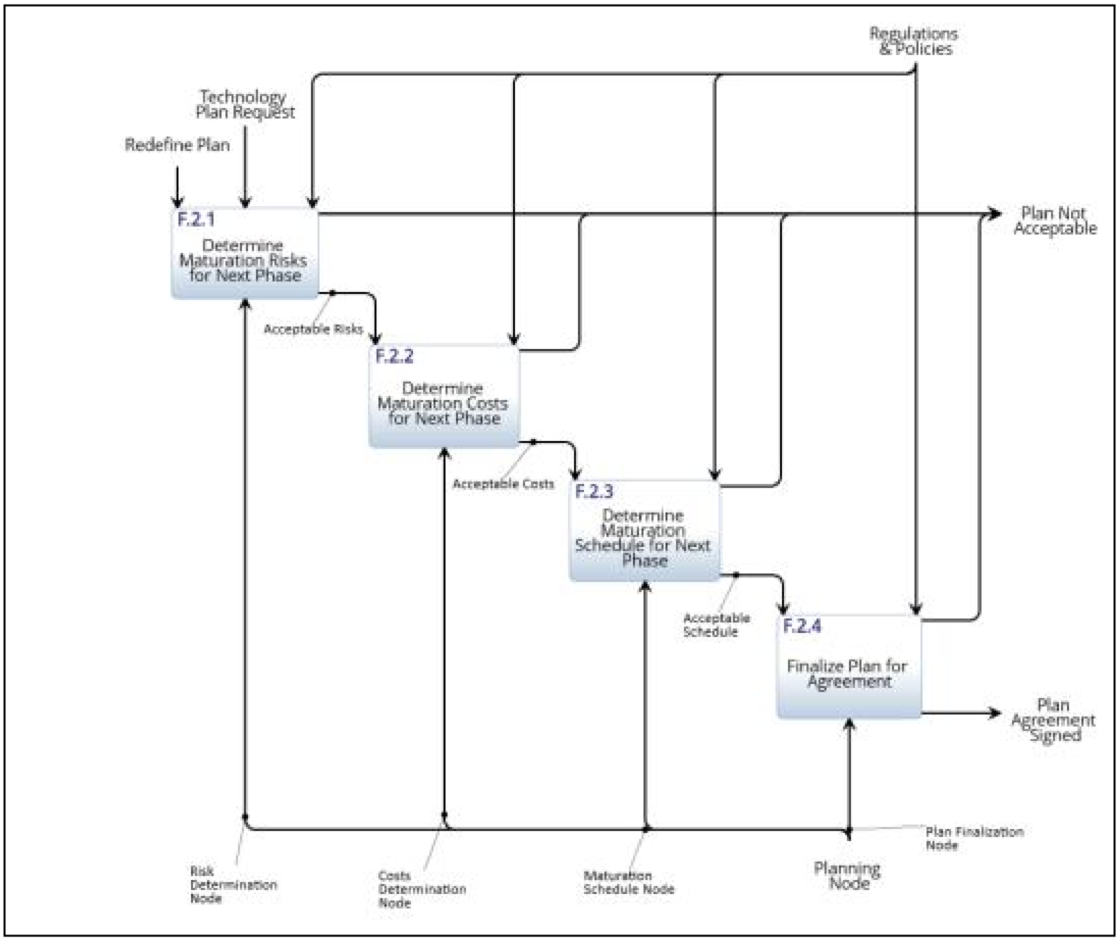


Figure 28. Produce Technology Development Plan IDEF0

This function was determined to be triggered by two possible controls: "Technology Development Plan" and "Redefine Plan." "Redefine Plan" indicates a subsequent unsuccessful iteration of the technology development plan. This control indicates that an assessment of the technology at one of the gated technology reviews revealed a lack of progress in the technology maturation. If the project is deemed to have potential to continue the maturation process, the "Redefine Plan" control will be sent back into the Produce Technology Development Plan function to trigger action.

The first sub-function, Determine Maturation Risks for Next Phase, was determined to perform an assessment of the current risks that have been defined for the technological development identifies any anticipated or projected risks for technological maturation. Two possible outputs were decided for this function: “Plan Not Acceptable” and “Acceptable Risks.” If the risks are deemed to be too great to overcome or if the identified mitigation procedures are insufficient to ensure success, the output “Plan Not Acceptable” is sent as a control to trigger activity within the Technology Maturation Closeout function. If the risks are deemed acceptable by all parties involved, then the output “Acceptable Risks” is sent as a control to trigger activity within the Determine Maturation Costs for Next Phase sub-function.

Determine Maturation Costs for Next Phase was determined to serve to produce a cost estimate for maturing the technology. This estimate is an approximation of the cost of the project as a whole. The cost estimate will have identifiable component values and use established methods, valid data, and will be based on what is known at the time of estimation. There are two possible outputs for this sub-function: “Plan Not Acceptable” and “Acceptable Costs.” An output of “Plan Not Acceptable” will be sent as a control and trigger activity internal to the Technology Maturation Closeout function. If it is determined that the plan and technology are supportable from a cost perspective, an output of “Acceptable Costs” will be sent as a control to trigger activity internal to the Determine Maturation Schedule for Next Phase sub-function.

Determine Maturation Schedule for Next Phase serves to produce a schedule estimate for maturing the technology from its current state to the next TRL (i.e. TRL 4 to TRL 5). The schedule estimation will include, but is not limited to, deliverable identification and timelines, project planning at all stages, gated review activities, design/build/test iterations, and transition artifact compilation. There are two possible outputs for this sub-function: “Plan Not Acceptable” and “Acceptable Schedule.” An output of “Plan Not Acceptable” will be sent as a control to trigger activity internal to the Technology Maturation Closeout function. “Acceptable Schedule” will provide notification that the technology possesses the attributes, from a schedule perspective, to be feasible within the constraints of the system. This output is then sent as a control to trigger activity internal to the Finalize Plan for Agreement sub-function.

Finalize Plan for Agreement serves to produce the finalized Technology Development Plan for maturing the technology. There are two possible outputs for this sub-function: “Plan Not Acceptable” and “Plan Agreement Signed.” An output of “Plan Not Acceptable” will be sent as a control to initiate activity internal to the Technology Maturation Closeout function. Once the plan is accepted and signed, the “Plan Agreement Signed” output is sent as a control to initiate activity within the Mature Technology function.

(3) Mature Technology. Mature Technology is decomposed into four sub-functions: Design Prototypes, Build Prototypes, Demonstrate Prototype in Simulated Environment, and Demonstrate Prototype in Operational Environment, as shown in Figure 29. Each sub-function of Mature Technology will be discussed in the following paragraphs along with inclusion of the graphical representation of the interactions.

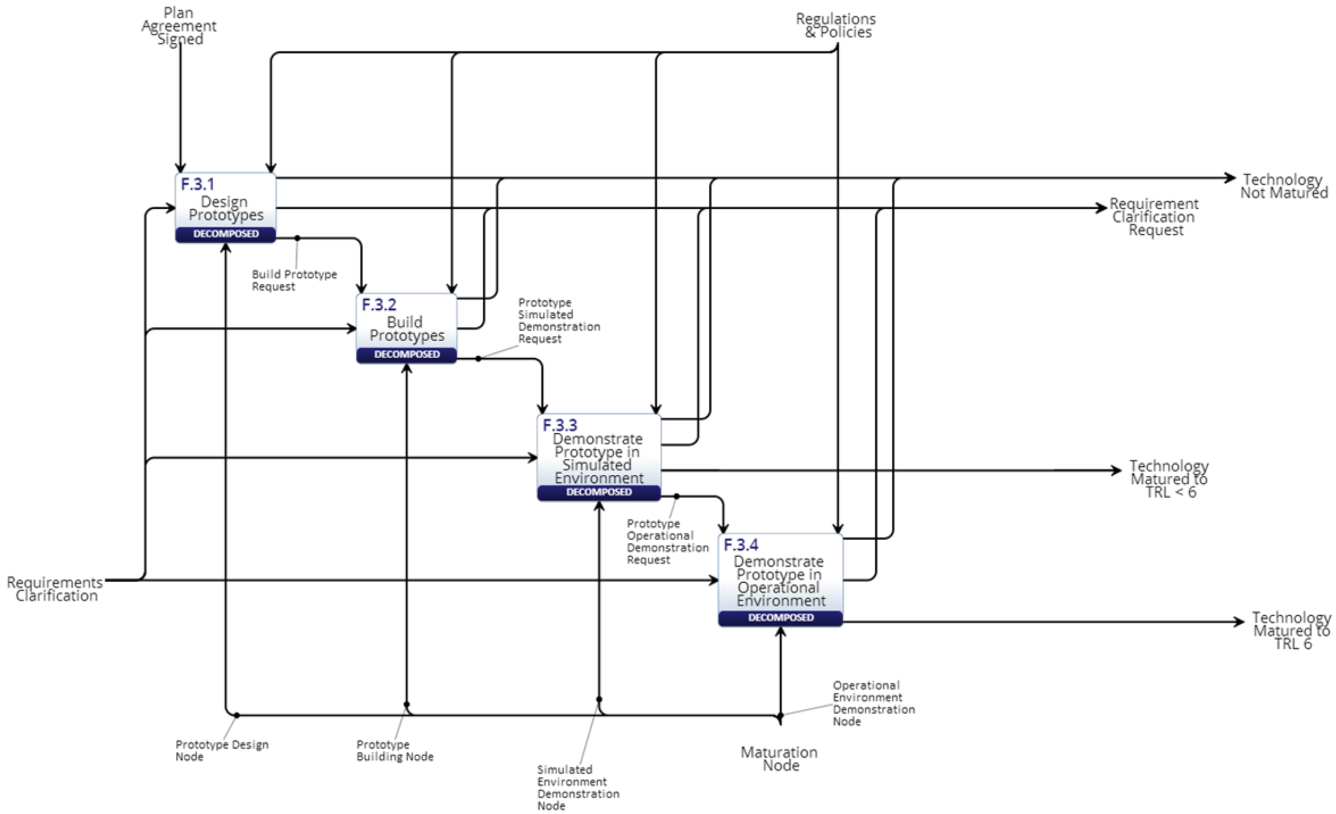


Figure 29. Mature Technology IDEF0

After the Technology Development Plan Agreement is signed, it becomes the control that triggers initiation of the first function inside Mature Technology: Design Prototypes. The Design Prototypes function, shown in Figure 30, is decomposed by nine sub-functions: Define System Boundary, Derive System Threads, Derive Component Hierarchy, Allocate Behavior to Components, Perform Modeling and Simulations, Perform Effectiveness and Feasibility Analysis, Select Design, Define Resources, Error Detection and Recovery, and Generate Documentation and Specifications.

The purpose of this function was to design a system representative prototype in accordance with the Technology Development Plan. The nine sub-functions are conducted iteratively and recursively to ensure the prototype design captures the necessary aspects of the user requirements as well as the steps outlined in the Technology Development Plan. The mechanism for the Design Prototypes function was designated as "Prototype Design Node." The Prototype Design Node will facilitate execution of the activities within the function. Design Prototypes encompasses many critical activities central to the success of the system.

This function provided: system boundary identification, system thread identification, component hierarchy identification, allocates behavior to individual functions, performs and evaluates modeling and simulation, analyzes the modeling and simulation results to select an optimized design solution, and finally provided a detailed design for building the prototype. The final output for the Design Prototypes function was determined to be the "Build Prototype Request." This request served as one of the controls and primary trigger for the follow on Build Prototypes function.

The Build Prototypes Function, shown in Figure 31, was decomposed into four sub-functions: Build Prototype Hardware, Build Prototype Software, Integrate Prototype Components, and Perform Component Integration Testing. This group of system functions purpose was to build the hardware and software in parallel and integrate the components into a testable prototype.

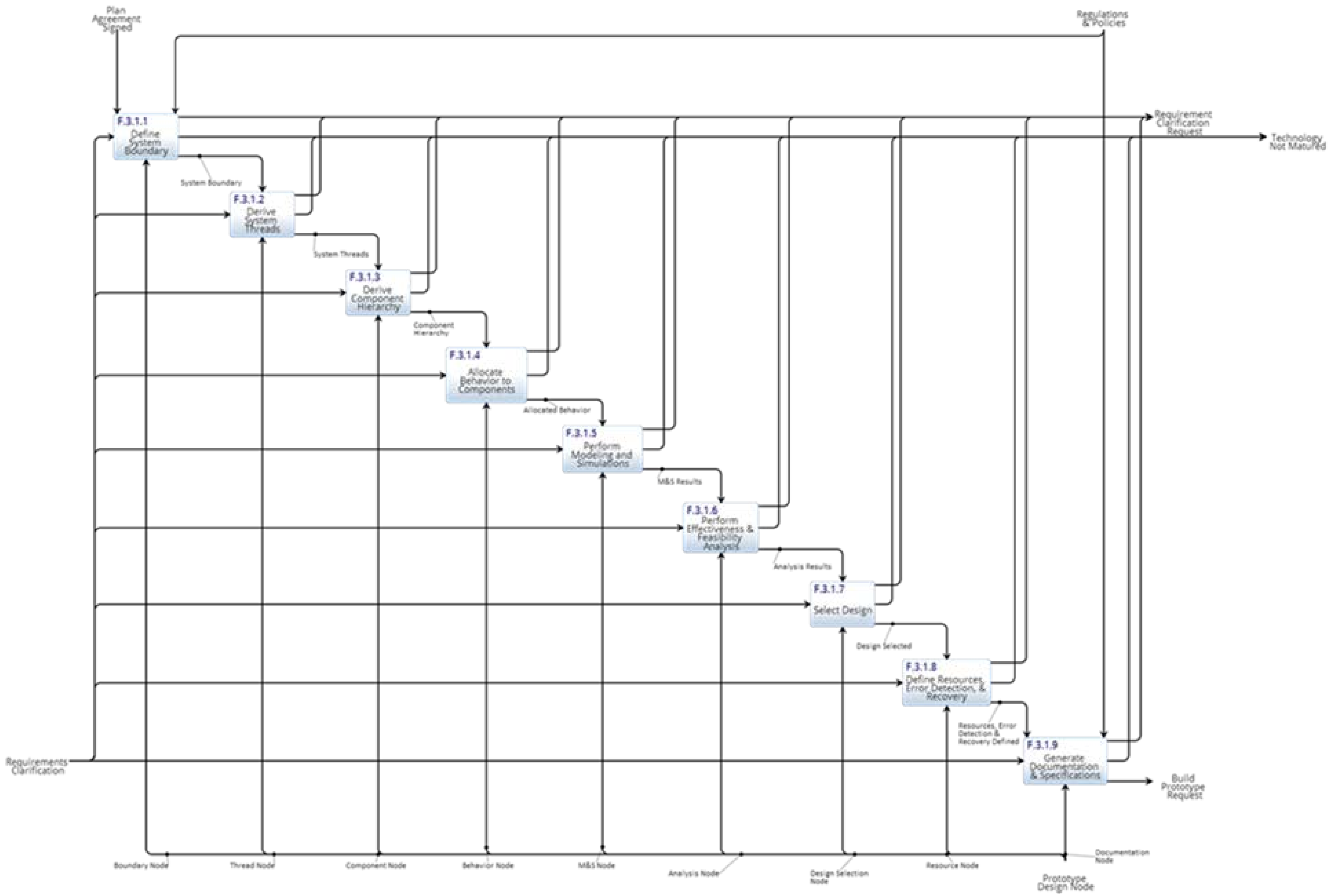


Figure 30. Design Prototype IDEF0

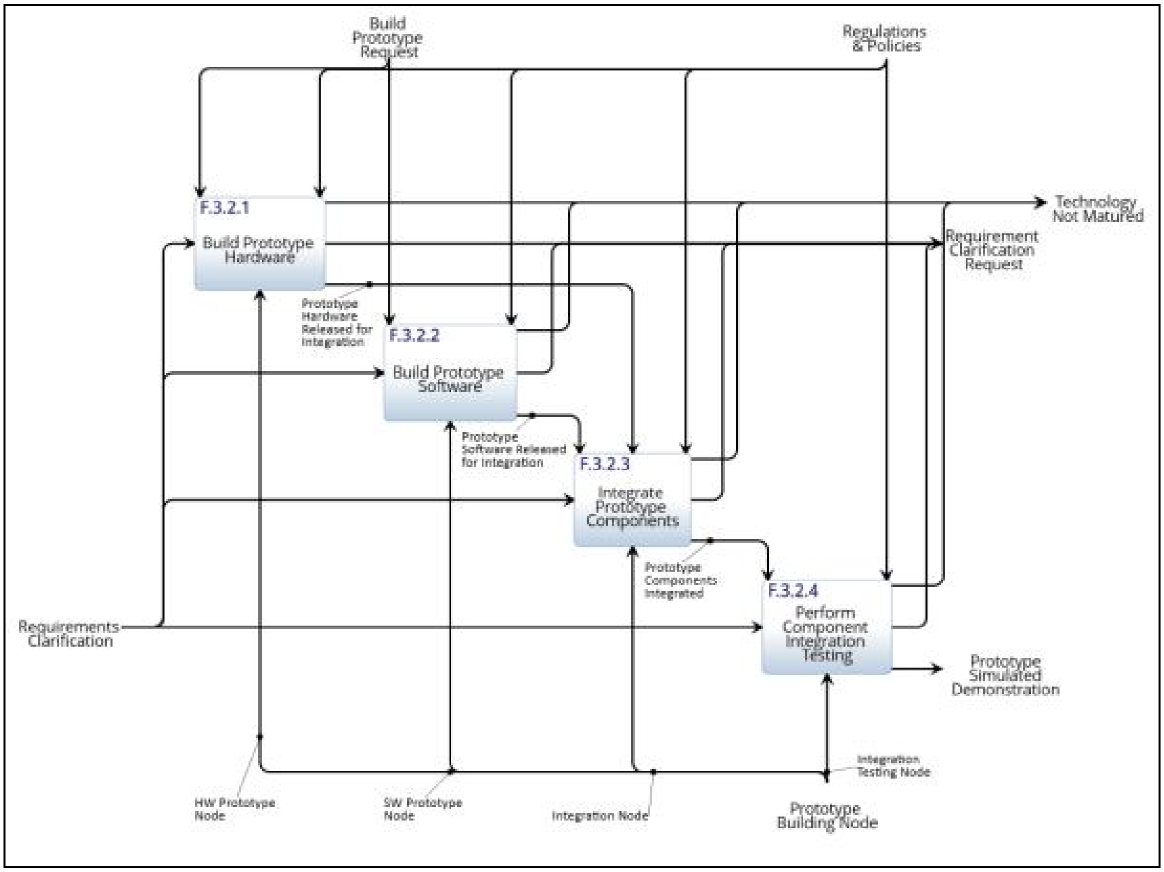


Figure 31. Build Prototypes IDEF0

The "Build Prototype Request" was determined to serve as a control for both Build Prototype Hardware and Build Prototype Software. This control was determined to trigger initiation of the activities within these two functions to begin a parallel effort of constructing the physical hardware and software components of the prototype. "Regulations and Policies" also served as a perpetual control on each function. The "Regulations and Policies" triggered any changes that are required should acquisition strategies or DOD directives be levied on the acquisition development community.

Upon conclusion of the activities within the parallel functions of Build Prototype Hardware and Build Prototype Software, three possible outputs can occur. If, during the course of the activities performed while building prototype hardware/software, a need should arise for requirements clarification, a request will be sent to the end user. If hardware and software development were successful, outputs are generated consisting of "Prototype Hardware Released for Integration" and "Prototype Software Released for Integration," respectively. Conversely, if it is determined during the course of building the prototype hardware and software that the technology cannot be matured, an output of "Technology Not Matured" will be generated and sent, as a control, to trigger the Redefine/Terminate Program function.

As shown in Figure 31, Integrate Prototype Components was determined to trigger the outputs from the Build Prototype Hardware and Build Prototype Software functions. These outputs serve as controls into the function and trigger the integration activities performed relative to building the prototype.

The integration activities can produce one of three separate outputs. If it is determined that the prototyping system needs further clarification of user requirements, a "Requirements Clarification Output" is sent to the Perform End User Activities function external to the system. Successful integration and assembly of the prototype hardware and software into a physical artifact will generate the output, "Prototype Components Integrated," that also serves as the control to trigger the Perform Component Integration Testing function. At any point during execution of the integration activities, an output of "Technology Not Matured" can be generated. This output is meant to provide a sanity check against the metrics defined in the Technology Development Plan. If it is determined that the technology cannot be matured within project constraints, this output is sent as a control, to trigger project redefinition or termination.

The Demonstrate Prototype in Simulated Environment function, in Figure 32, is decomposed by three sub-functions: Model Simulated Environment, Run Prototype in Simulated Environment, and Evaluate Results. This function serves to perform the demonstration of the prototype in a simulated environment and evaluate the results produced by the unit under test.

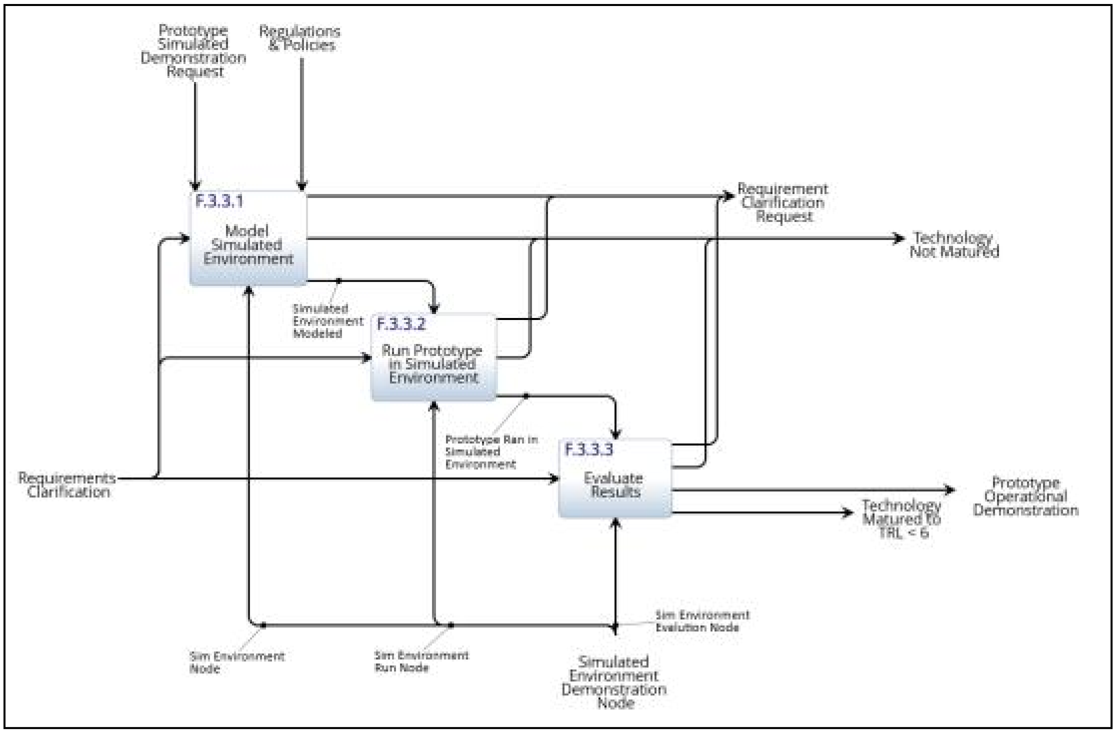


Figure 32. Demonstrate Prototype in Simulated Environment IDEF0

Successful development of prototype hardware/software, component integration, and component integration testing during the Build Prototypes sub-function was determine to culminate in the output, “Prototype Simulated Demonstration Request.” This output served as the primary control and trigger for initiation of the Demonstrate Prototype in Simulated Environment function. The first sub-function is Model Simulated Environment. The activities within this sub-function utilize information provided the user, for example, requirements, capability requests, and operational use cases, to develop an approved simulated environment for prototype testing. Based on the team research and analysis, there were three possible outputs for the Model Simulated Environment function. If it is determined that the prototyping system needs further clarification of user requirements, a Requirements Clarification Output is sent to the Perform End User Activities function external to the system. At any point during development of the simulated environment, an output of Technology Not Matured can be generated. This output is meant to provide a sanity check against the user defined metrics located within the Technology Development Plan. If is it determined that the technology cannot be matured within project constraints, this output is sent to, and serves as, a control to trigger project redefinition or termination. The primary output of the function is Simulated Environment Modeled. This output serves as a control and primary trigger for the Run Prototype in Simulated Environment function. After receipt of the “Simulated Environment Modeled” trigger is received by the Run Prototype in Simulated Environment function, activities are initiated for demonstration of the prototype. The demonstration at this stage of development is meant to show a technological maturity consistent with the metrics for a TRL 5 technology. There are three possible outputs for Run Prototype in Simulated Environment: “Requirements Clarification Request,” “Technology Not Matured,” and “Prototype Ran in Simulated Environment.” “Prototype Ran in Simulated Environment” is the desired output and represents the determination that the technology is feasible and supportable to advance into the follow on stages of development. This output serves as the primary control for the Evaluate Results function and triggers analysis of the results produced during simulated testing of the prototype.

Evaluate Results is triggered by the “Prototype Ran in Simulated Environment” control. The activities within this sub-function serve to evaluate the results and generated data produced by the prototype under test within the simulated environment. There are four possible outputs for the Evaluate Results function: “Requirements Clarification Request,” “Technology Not Matured,” “Prototype Operational Demonstration,” and “Technology Matured to TRL 6.”

The intent of demonstrating the prototype in a simulated environment is to advance technological maturity to TRL 5. It is conceivable, however, that a technology could be deemed TRL 6 after demonstration within the simulated environment. If this case presents itself, an output of “Technology Matured to TRL 6” will be generated and sent as a control to trigger activities within the Transition Technology function. The primary output of the Run Prototype in Simulated Environment function is the “Prototype Operational Demonstration” request. This output will trigger initiation of the Demonstrate Prototype in Operational Environment function.

Demonstrate Prototype in Operational Environment, shown in Figure 33, is decomposed by three sub-functions: Validate Operational Environment, Demonstrate

Prototype in Operational Environment, and Evaluate Results. This function serves to perform the demonstration of the prototype in an operational environment relative to the intended users of the system.

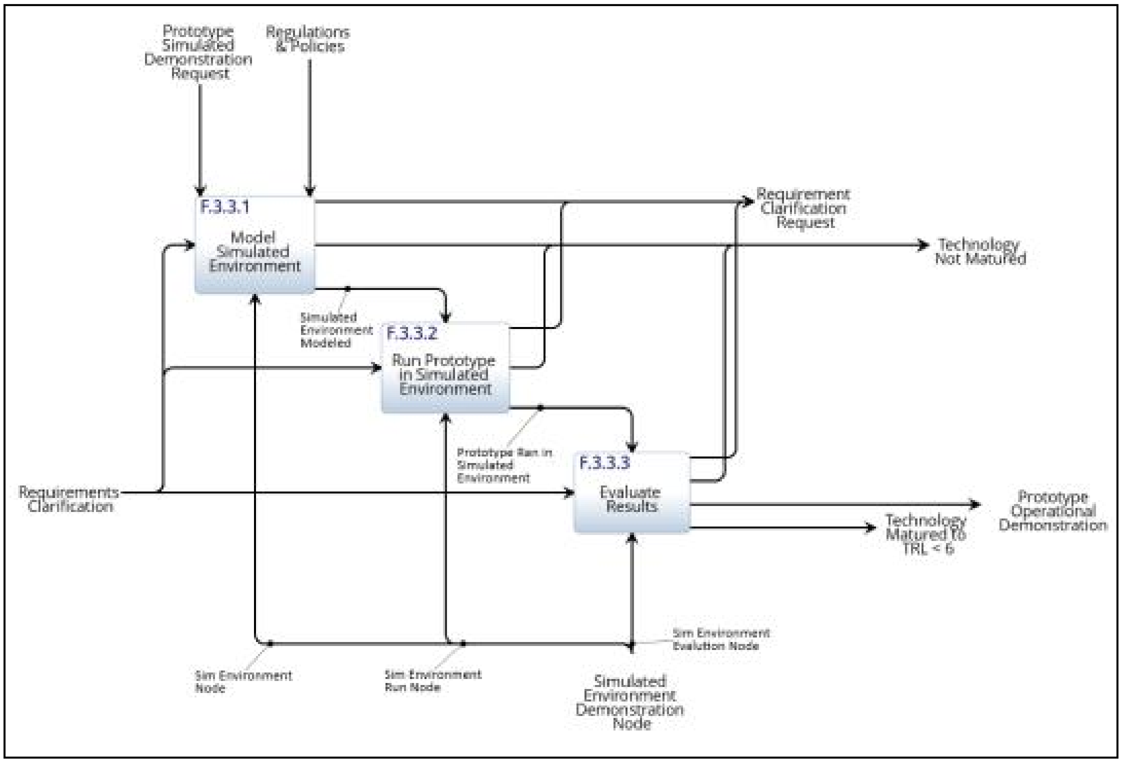


Figure 33. Demonstrate Prototype in Operational Environment

Successful demonstration of the prototype in the simulated environment would generate an output, and subsequent control titled "Prototype Operational Demonstration Request," this will initiate activities within the Demonstrate Prototype in Operational Environment function.

Validate Operational Environment would utilize user requirements, operational reference scenarios, and use case validations from the user to validate the operational environment to which the technology under development would be tested. There are three possible outputs from the Validate Operational Environment function: "Requirements Clarification Request," Technology Not Matured," and "Operational Environment Validated."

The primary and intended output of this sub-function is “Operational Environment Validated.” In order to meet the criteria for TRL 6, a technology must be demonstrated in the intended operational environment. This represents a major step in the readiness of the demonstrated technology. The operational environment must validate the expectations of the user and prove that the system can be advanced into an operational system.

Once the “Operational Environment Validated” control is received by Demonstrate Prototype in Operational Environment, activities are initiated to perform the prototype demonstration. There are three possible outputs for this function: “Requirements Clarification Request,” “Technology Not Mature,” and “Demonstration in Operational Environment Complete.”

The primary and intended output of the Demonstrate Prototype in Operational Environment function is “Demonstration in Operational Environment Complete.” This output serves as the control to initiate activities within the Evaluate Results function. This function serves to evaluate the results and data produced during operational demonstration. This evaluation is meant to measure how the test compared with expectations and intended results, identify any problems encountered, and identify plans and options to resolve the problems before advancing into the next stage.

There are three possible outputs for the Evaluate Results sub-function: “Requirements Clarification Request,” “Technology Not Matured,” and “Technology Matured to TRL 6.” The primary output, “Technology Matured to TRL 6,” is the ultimate goal of the activities within the system as a whole. If, after successful evaluation of the operational demonstration results, the technology is deemed to be matured to TRL 6, this output will serve as the control to initiate activities within the Transition Technology Function.

Transition Technology, shown in Figure 34, is decomposed by three sub-functions: Finalize Technology Transition Artifacts, Perform Technology Readiness Assessment for Transition, and Transition Technology Artifacts. The collection of activities internal to the Transition Technology function serves to determine whether the technology under development has shown sufficient evidence to meet the criteria for transition to the customer.

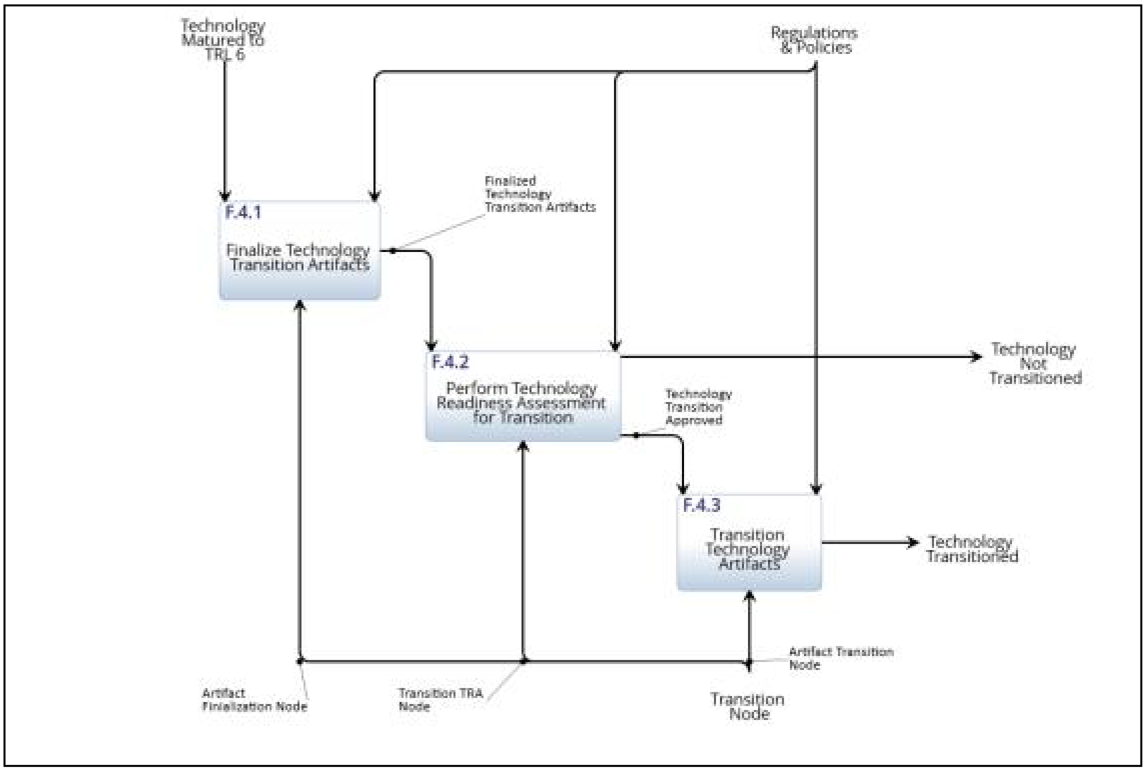


Figure 34. Transition Technology IDEF0

Successful maturation of the technology to TRL 6 generates an output from the Mature Technology function. This output becomes the control to initiate activities within the first Transition Technology sub-function, Finalize Technology Transition Artifacts. This sub-function serves to finalize the artifacts produced during system activities and is utilized to support the Transition Technology Readiness Assessment.

The output of this sub-function, “Finalized Technology Transition” becomes the primary control for Perform Technology Readiness Assessment for Transition. This control initiates the transition assessment of the technology under development in order to ensure the exit criteria specified in the Technology Development Plan has been achieved. This function has two possible outputs: “Technology Not Matured,” and “Technology Transition Approved.”

The primary output, “Technology Transition Approved,” validates that the technology has achieved TRL 6 or greater and is sufficiently mature to transition to the customer. This output becomes the primary control to initiate activities within the Transition Technology Artifacts sub-function.

Upon receipt of the approved Transition Technology Readiness Assessment, the Transition Technology Artifacts function is triggered to execute. This function will map the progress of the technology based on its advancement through the specifications of the Technology Development Plan. This is a collaborative function that synergizes the efforts of the system with the expectations of the customer. It will serve to ensure, to all stakeholders, that the technology that entered the system at TRL 4 has, in fact, been developed, tested, and demonstrated in an operational environment and has achieved TRL 6.

The primary and intended output of the Transition Technology Artifacts function is “Technology Transitioned.” This output provides validation that the technology under development has matured to TRL 6 and is approved for transition to the customer. This output will serve as the control for the Technology Maturation Closeout that triggers the associated activities within that function. The output titled, “Technology Not Transitioned,” is also a potential should it be deemed that the technology under development has not sufficiently achieved the required level of maturity for transition. This output will serve as a control to the Redefine/Terminate Program function and trigger program reevaluation.

Redefine/Terminate Program, shown in Figure 35, is decomposed by three sub-functions: Program Determination, Redefine Program Plan, and Capture Issue Metrics. This function serves to determine whether the Technology Development Plan for the specific technology under development should be redefined or if the technology lacks sufficient merit to continue and requires termination.

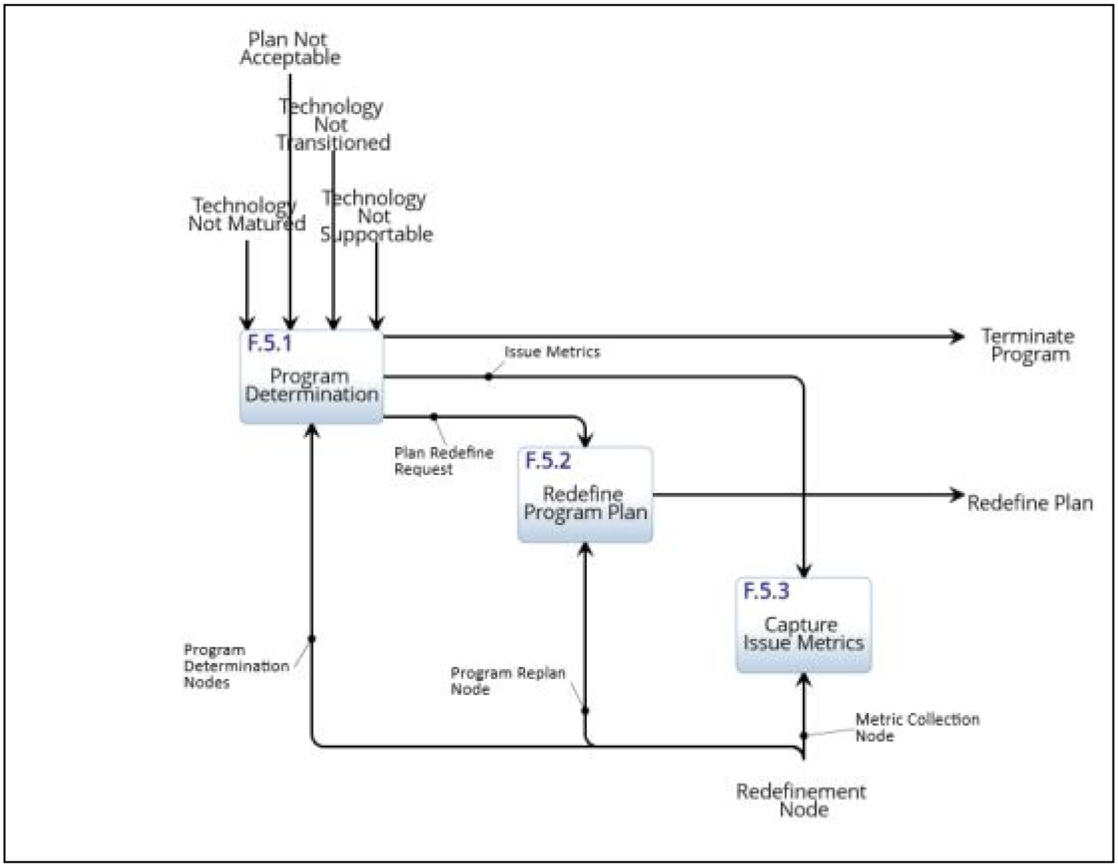


Figure 35. Redefine/Terminate Program IDEF0

As shown in Figure 35, the Redefine/Terminate Program function can receive a control from any one of the preceding functions to trigger an assessment of the overall development program strategy. Upon receipt of any of the controls titled, "Technology Not Supportable," "Technology Not Transition," "Plan Not Acceptable," or "Technology Not Matured, the Program Determination sub-function is initiated. This function serves as an opportunity for the program status to be assessed in order to determine whether the program should be redefined or terminated.

There are three possible outputs for the Program Determination sub-function: "Terminate Program," "Issue Metrics," and "Plan Redefine Request." "Terminate Program" executes activities to terminate the program by serving as a control to trigger activities within the Technology Maturation Closeout function. "Issue Metrics" executes activities, which capture programmatic and technological metrics in order to identify issues that may have spurred issues during technology development. This output serves as the control to trigger activities within the Capture Issue Metrics sub-function. "Plan Redefine Request" executes the activities necessary to determine if the technology is still viable should the plan be redefined. This output becomes the primary control to trigger activities within the Redefine Program Plan sub-function.

Upon receipt of the "Plan Redefine Request," the Redefine Program Plan sub-function is initiated. This sub-function serves to notify the Produce Technology Development Plan function that the program requires a redefinition and what specific information served to hinder the technology within the context of the development cycle. As stated, the primary output for this sub-function is "Redefine Plan." Once this output reaches the Produce Technology Development Plan function, activities are initiated to redefine the direction of the project.

The Capture Issue Metrics sub-function receives the output "Issue Metrics" as a control to initiate its internal activities. The observed and recorded metrics for faults or failures within the system are used to support the identification of process improvement areas. There are many issues that, with proper planning and redefinition, can be mitigated. The primary output, "Running Issue Metrics," defines the issues and mitigation steps that will be necessary when redefining the program plan. This output is sent to Produce Technology Development Plan and serves as a trigger to initiate activities internal to that function.

The Technology Maturation Closeout function is a standalone functional entity that serves as the final closeout operation of the DOD Prototyping System. Regardless of whether technology development has been successful or if it was determined that project termination was the appropriate action, the closeout function will execute the activities necessary to provide the response to the customer. All artifacts developed during system activities as well as transition documentation, etc, will be transitioned to the customer through this function.

There are two primary controls that initiate activity within this function: "Technology Transitioned" and "Terminate Program." Each of these controls will trigger similar activities within this sub-function. The primary output from the sub-function is the “Service Response.” This output is sent directly to the external Perform Customer Activities function.

4. Functional Analysis Summary

In the Functional Analysis section, top-level TDS functions and sub-functions were identified, along with clarification of the ICOMs, which represent relationships and interactions between the functions and sub-functions. The functions of the TDS were decomposed to the level sufficient to explain the functional implementation of the system concept. The functional analysis is an important step, and the structural backbone for development of the functional and input-output requirements in the following section. (Blanchard and Fabrycky 2011)

B. REQUIREMENTS

Requirements Analysis (RA) is the process of reviewing, assessing, prioritizing, and balancing all stakeholders and derived requirements including the constraints. The goal of RA is requirements allocation to transform those requirements into a functional and technical view of a system description capable of meeting the customer’s needs and objectives. (INCOSE 2010) Requirements management is a key element of the systems engineering processes. Requirements management is the identification, derivation, allocation, and control in a consistent, traceable, associative, verifiable manner of all the system functions, attributes, interfaces, and verification methods that a system must meet including customer, derived (internal), and specialty engineering needs (Buede 2009).

The methodology for requirements management was determined, based on the work of Dennis M. Buede, Benjamin S. Blanchard and Wolter J. Fabrycky. A top-level system requirement was developed along with Input-Output, Interface, Constraint, Functional and Non-Functional Requirements. The requirements were traced to ensure applicability within an operational context and have traceability to some stakeholder value (2009; 2011).

The requirements for the TDS were determined by analyzing the problem, reviewing stakeholder needs, and reviewing the functional analysis. Without a specific customer or user for this capstone project to provide feedback, the team relied upon extensive research to develop relevant requirements for the TDS. Requirements analysis typically involves frequent communication with the system users to determine specific expectations and to resolve any ambiguity in the requirements. The lack of a specific user representative forced the need to rely on prior research and decision papers detailing the issues that plague the current technology development and prototyping efforts in DOD acquisition. System requirements were developed, analyzed, and applied in a team environment such that the identified gaps in the TMRR phase of DOD acquisition were sufficiently addressed by the final system concept. The final system concept to system requirements match up provides standards and measurement tools for determining success of the system design (Buede 2009).

The top-level system requirement was derived from the problem statement, stakeholder needs, and concept of operations. Once the top-level system requirement was developed, it enabled the identification of system functional requirements that defined the activities or functions that the system must perform. Identification of the system level requirements provided the building blocks necessary to transform the functional, input/output, performance, and non-functional requirements into a coherent description of systems functions known as the functional architecture. This was accomplished by arranging the functions in logical sequences, decomposing higher-level functions, and allocating performance from higher-to lower-level functions. The tools that were used to perform this analysis include functional flow block diagrams and IDEF0 illustrations. The functional architecture provided a description of what the system must do, but in terms of functional and performance parameters, rather than a physical description. Functional Analysis and Allocation facilitated traceability from requirements to the system solution (DOD Systems Management College 2013).

1. Top-level System Requirement

The top-level system requirement, derived from the problem definition, stakeholder needs, and concept of operations, was created to capture the high level intent and demands of the system. The top-level system requirement is presented in Table 4.

Table 4. Top-level System Requirement

Number	Top Level System Requirement
1	The system shall provide DoD Acquisition Authorities with processes to perform prototyping between MS A and MS B in order to mature technological capabilities.

2. Functional Requirements

The functional requirements stemmed from the functions developed to support the top-level system requirement. These functional requirements for the TDS are presented in Table 5.

Table 5. Functional Requirements

Number	Functional Requirements
2.1	The system shall assess project feasibility.
2.1.1	The system shall perform a technology readiness assessment.
2.1.2	The system shall assess technology feasibility.
2.1.3	The system shall assess programmatic feasibility.
2.2	The system shall produce a technology development plan
2.2.1	The system shall determine maturation risks for the next phase.
2.2.2	The system shall determine maturation costs for the next phase.
2.2.3	The system shall determine maturation schedule for the next phase.
2.2.4	The system shall finalize a technology development plan for agreement.

Table 5. Functional Requirements

Number	Functional Requirements
2.3	The system shall mature technology
2.3.1	The system shall design prototypes.
2.3.1.1	The system shall define the prototype system boundary.
2.3.1.2	The system shall derive prototype system threads.
2.3.1.3	The system shall derive prototype component hierarchy.
2.3.1.4	The system shall allocate behavior to prototype components.
2.3.1.5	The system shall perform modeling.
2.3.1.6	The system shall perform simulations.
2.3.1.7	The system shall perform effectiveness analysis.
2.3.1.8	The system shall perform feasibility analysis.
2.3.1.9	The system shall select a prototype design.
2.3.1.10	The system shall define prototype resources.
2.3.1.11	The system shall define error detection.
2.3.1.12	The system shall define recovery.
2.3.1.13	The system shall generate documentation.
2.3.1.14	The system shall generate specifications.
2.3.2	The system shall build prototypes.
2.3.2.1	The system shall build prototype hardware.
2.3.2.2	The system shall build prototype software.
2.3.2.3	The system shall integrate prototype components.
2.3.2.4	The system shall perform component integration testing
2.3.3	The system shall demonstrate the prototype in a simulated environment
2.3.3.1	The system shall model a simulated environment.
2.3.3.2	The system shall run the prototype in a simulated environment.
2.3.3.3	The system shall evaluate the results of the simulation.
2.3.4	The system shall demonstrate the prototype in an operational environment.
2.3.4.1	The system shall validate the operational environment.
2.3.4.2	The system shall demonstrate the prototype in an operational environment.
2.3.4.3	The system shall evaluate the demonstration results.
2.4	The system shall transition technology.
2.4.1	The system shall finalize technology transition artifacts.
2.4.2	The system shall perform a technology readiness assessment for
2.4.3	The system shall transition technology artifacts.
2.5	The system redefine or terminate the program.
2.5.1	The system shall make a program determination.
2.5.2	The system shall redefine the technology development plan.
2.5.3	The system shall capture issue metrics.
2.6	The system shall perform technology maturation closeout.

3. Input-Output Requirements

The input and output requirements were derived from the high level TDS design operational concept description and system life cycle. As shown in Figure 36, the TDS ICOMs are limited at a high level. There are many lower level inputs and outputs for each function that were not decomposed for the systems level requirements analysis. The controls, or triggers, were also considered to be inputs and treated as input requirements.

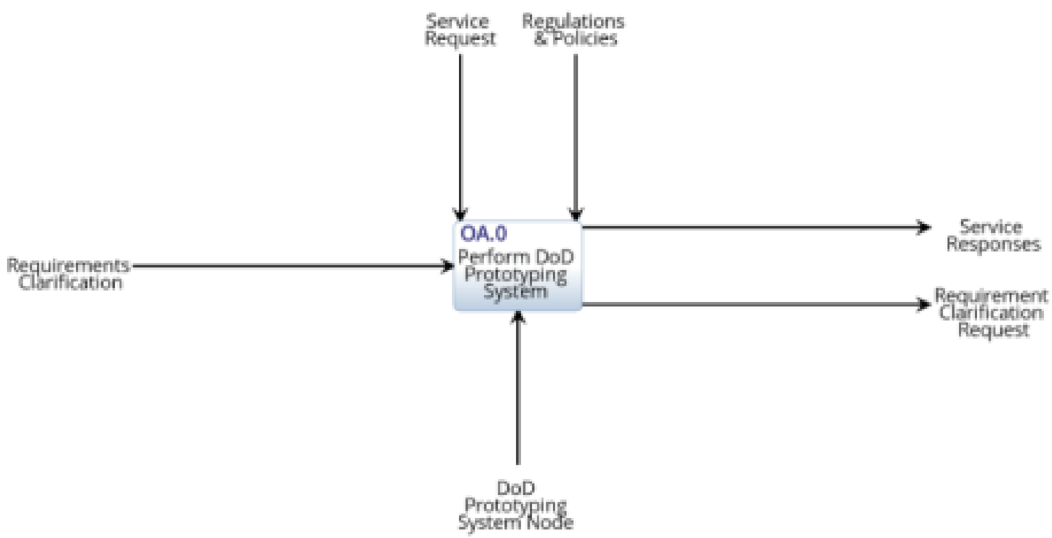


Figure 36. TDS A0 Diagram

The input and output requirements for the TDS are summarized in Table 6.

Table 6. Input and Output Requirements

Number	Input-Output Requirements
3.1	The system shall accept service requests.
3.2	The system shall accept requirements clarifications.
3.3	The system shall accept govermental regulations.
3.4	The system shall accept govermental policies.
3.5	The system shall produce service responses.
3.6	The system shall produce requirements clarification requests.

4. Non-Functional Requirements

According to Buede, performance requirements include the "ilities" and the non-functional characteristics of the entire system (Buede 2009). Users have implicit expectations about how well a system should work. These characteristics include how easy the system is to use and how reliable the system will be when attempting to repeat a process. Non-functional requirements are vast and not easily defined. Due to the academic constraints placed on this project, the non-functional requirements were limited to the two most precise "ilities" that were most important to the team; usability and repeatability.

A system's utility is characterized by its ability to be implemented and utilized by the stakeholders; therefore, these requirements were meant to ensure the system produces repeatable results, regardless of the project, and to enable the use of the TDS across all services of the DOD (Blanchard and Fabrycky 2011). Usability was identified as a non-functional requirement in order to address the factors that constitute the capacity of the process model to be understood, learned, and used by its intended users. Repeatability specifies the capability of the process model to maintain its performance over time by providing a set of activities that can be easily duplicated with multiple user groups. (Bahill and Gissing 1998) These requirements are presented in Table 7.

Table 7. Non-Functional Requirements

Number	Non-Functional Requirements
4.1	The system shall produce repeatable results.
4.2	The system shall be usable by all services in the DoD.

The first iteration of the requirements analysis included other performance requirements such as interoperability, safety, and tailorability. The interoperability requirement was initially included to ensure the TDS worked within the parameters of DOD regulations and acquisition laws. The safety requirement was intended to mitigate safety risks using the guidelines of MIL-STD-882. The intent of the tailorability requirement was to provide system flexibility to accommodate variations in scope, size, cost, and maturity of DOD acquisition projects.

These requirements were not included due to the difficulty in developing quality requirements. Quality requirements must be correct, feasible, unambiguous and verifiable. (Wiegers 1999) Correctly conveying the intent of the requirement without becoming infeasible proved to be an arduous task. The real difficulty came when trying to create unambiguous and verifiable requirements. Defining units of measurement for the interoperability, safety, and tailorability of a prototyping process required research that was outside the scope of the project. In addition, understanding the definition, classification and representation of the non-functional or performance requirements for the TDS added to the complexity of properly addressing the non-functional requirements (Glinz 2007). As described in the recommendations for future work, follow-on research should be performed to better understand how performance requirements can be applied to a process such as the TDS.

5. Interface Requirements

Interface requirements address total system performance, the attributes of the interface, and any system requirements meant to constrain interface design. The requirements define the external interfaces with other systems in terms of message format, content configuration characteristics, devices supported, protocols, speed, through put and response time. The interface should not change the items during the transmission process. (Buede 2009) Several interface requirements were considered for the TDS process under development. Most hard coded interface requirements were rejected due to the considerations given to the process to remain flexible and tailorable as high priority attribute.

Table 8. Interface Requirements

Number	Interface Requirement Requirements
5.1	The TDS system shall conform to all Regulations and Policies
5.2	The TDS system shall confirm receipt of all Service Request within one week

6. Constraint Requirements

When a new system's boundaries are defined with external inputs and outputs during system design the items outside of the boundaries cannot be changed and the items within the boundaries of the system are subject to change depending on the requirements. System context diagrams or other graphical representations can be used to visually show the data flows displaying the system's boundaries with system inputs and outputs displayed in the relevant contexts. (Buede 2009) The constraint requirements will be validated and verified throughout the system life cycle especially at the point at which physical architecture and design are being developed. (DOD Systems Management College 2013)

Table 9. Constraint Requirements

Number	Constraint Requirements
6.1	The TDS shall operate within the confines of the current DoD acquisition structure.
6.2	The TDS shall preform prototyping between Milestone A and Milestone B in the acquisition process.
6.3	The TDS shall accept technology of TRLof equal or greater than 4 and mature it to a TRL equal or greater than 6.
6.4	The TDS shall mature technology and capabilities, while rejecting technologies and capabilities that cannot meet the appropriate TRL
6.5	The TDS shall have review points to ensure the technology is maturing according to the plan and budget.
6.6	The TDS process can be performed by government organizations or contractor organizations.
6.7	A primitive need statement shall be provided for the technology entering the TDS.
6.8	The need for a prototype shall be provided to advance the development of the technology has been confirmed.
6.9	A program schedule shall be provided for the technology entering the TDS.

C. VALUE SYSTEM DESIGN

In order to assess the feasibility of the TDS, a model was created of the value system. A value system is an extension of the Stakeholder Analysis and Functional Analysis. According to Keeney, the value model takes each system function and identifies the objective of that function. The objective must be clear and measureable in order to be sure the intended need of the function is met. When an objective is determined, an evaluation measure is identified that provides a method for determining if the objective has been accomplished. These objectives are a reflection of the goals and needs of the system stakeholders, as identified in the Stakeholder Analysis. All of these components, functions, objectives, and evaluation measures are combined to create a value hierarchy diagram that represents the system's value system structure. (Keeney 1992)

1. Value Hierarchy

After completing the functional analysis and developing the functional hierarchy, the team utilized this information to develop a value hierarchy for the system. As depicted in Figure 37, the value hierarchy represents the functions, objectives, and metrics of the system that can be used to compare system concepts.

These functions represent all of the activities that comprise the development of a prototype; however, the focus of this project is the Perform DOD Prototyping Activities function and its sub-functions. The top-level prototyping function was further decomposed into sub-functions Assess Feasibility, Produce Technology Development Plan, Mature Technology, Transition Technology, Redefine/Terminate Program, and Technology Maturation Closeout. Assess Feasibility is the evaluation and analysis of information needed to determine if a prototype can be developed. Produce Technology Development Plan documents the steps required to create a prototype. Mature Technology defines the activities defined in the Technology Development Plan. The output of this function is a prototype that has reached a TRL of 5 or 6. Transition Technology describes the activities required for delivering the prototype to the customer. Redefine/Terminate Program defines the activities associated with the reevaluation or termination of a prototype. Technology Maturation Closeout comprises the activities required to close out the prototyping phase and send a service response to the customer.

In addition to modeling the functions, the non-functional attributes for prototype development were also identified and defined. Repeatability requires that the prototyping process provide similar results for any development program. Usability states that the process must fit the needs of all DOD services.

After establishing the functions and non-functional attributes required for prototype development, the objectives and evaluation measures were defined. The evaluation measures are discussed in more detail in the next section.

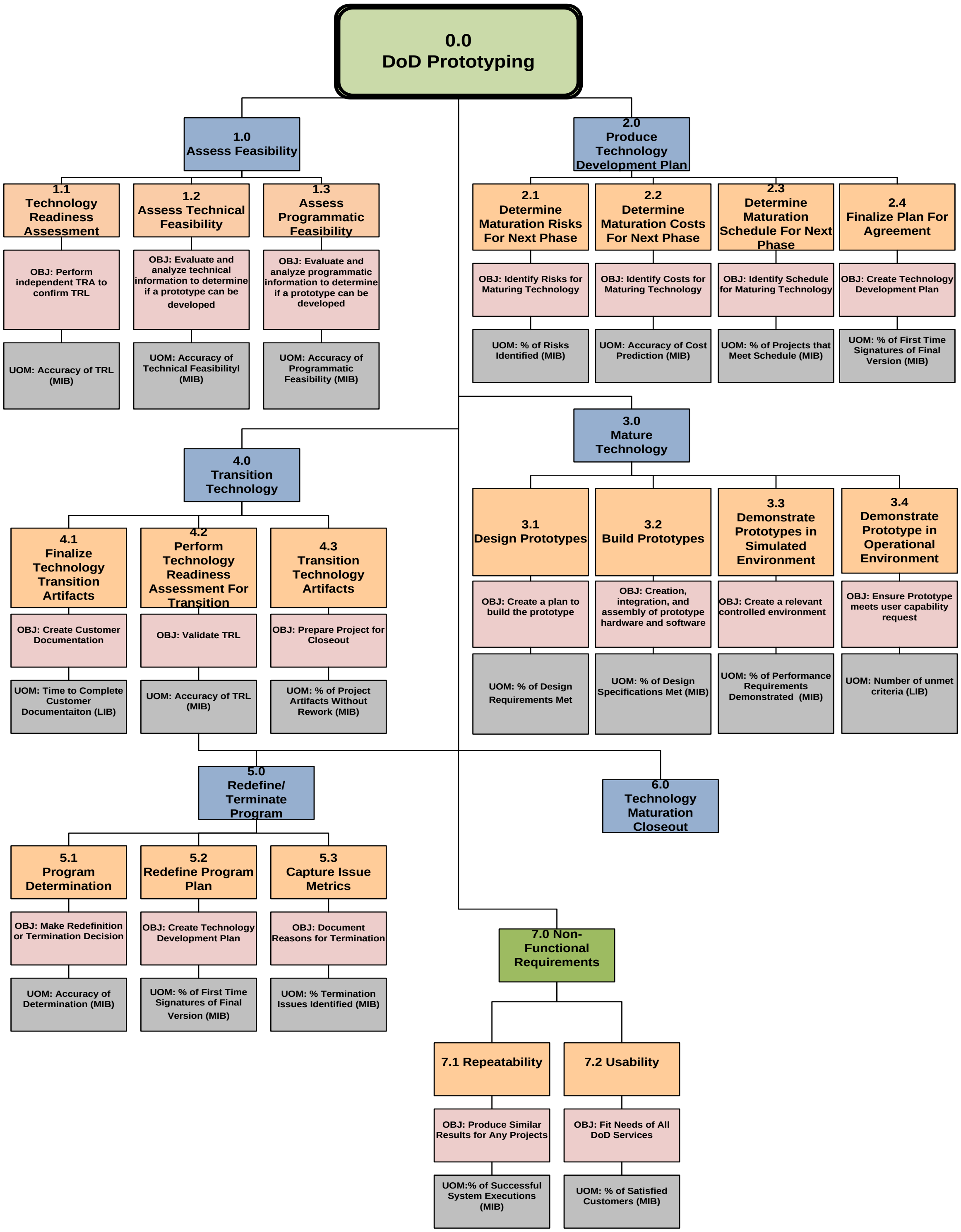


Figure 37. Value Hierarchy

The value hierarchy identified the objections and evaluation measures for twenty-two functions and non-functional requirements.

2. Evaluation Measures

According to Keeney, evaluation measures (EMs) are specific measures used to evaluate the level at which the design met the customer's needs. The evaluation measures that were utilized were defined as being direct or proxy. Direct measures look at the attainment of the objective in question. Proxy measures focus more on an associated objective. Evaluation measures can be further defined as natural or constructed. A natural measure is one that has universal application and clear and concise meaning, while constructed measures are developed from a combination of measures. (Keeney 1992) Table 10 contains a list of all the system evaluation measures along with their classification. The measure column indicates, More is Better (MIB) or Less is Better (LIB) in the table. For MIB, the greater the value on a measure the better for the stakeholder, while the LIB is the opposite. The objective is the stakeholder's desire, while the threshold is the minimum acceptable value or measure.

Table 10. Evaluation Measures

Functional Trace	**Evaluation Measure**	**Measure**	**Units**	**Classification**	**Threshold**	**Objective**
1.1	Accuracy of TRL	MIB	%	Proxy/Constructed	95%	100%
1.2	Accuracy of Technical Feasibility	MIB	%	Proxy/Constructed	95%	100%
1.3	Accuracy of Programmatic Feasibility	MIB	%	Proxy/Constructed	95%	100%
2.1	% of Risks Identified	MIB	%	Direct/Constructed	95%	100%
2.2	Accuracy of Cost Prediction	MIB	%	Proxy/Constructed	80%	90%
2.3	% of Projects that Meet Schedule	MIB	%	Proxy/Constructed	95%	100%

Functional Trace	Evaluation Measure	Measure	Units	Classification	Threshold	Objective
2.4	% of First Time Signatures of Final Version	MIB	%	Direct/Natural	97%	100%
3.1	% of Design Requirements Met	MIB	%	Direct/Natural	80%	100%
3.2	% Design Specifications Met	MIB	%	Direct/Natural	80%	100%
3.3	% of Performance Requirements Demonstrated	MIB	%	Direct/Natural	90%	100%
3.4	% of Operational Performance Criteria Demonstrated	MIB	%	Direct/Natural	90%	100%
4.1	Time to Complete Customer Documentation	LIB	Time	Direct/Natural	6 Weeks	3 Weeks
4.2	Accuracy of TRL	MIB	TRL	Direct/Natural	95%	100%
4.3	% of Project Artifacts Without Rework	MIB	%	Direct/Natural	95%	100%
5.1	Accuracy of Determination	MIB	%	Direct/Natural	95%	100%
5.2	% of First Time Signatures of Final Version	MIB	%	Direct/Natural	97%	100%
5.3	% of Termination Issues Identified	MIB	%	Proxy/Constructed	95%	100%
7.1	% of Successful System Executions	MIB	%	Direct/Natural	95%	100%
7.2	% of Satisfied Customers	MIB	%	Proxy/Natural	95%	100%

Each of the EMs in Table 10 were linked to specific functions and were intended to measure the effectiveness of a solution at meeting the objective of the function. In order to make the value hierarchy as complete as possible, the evaluation measures at the lower layer of the hierarchy, when taken together as a group, were developed to adequately cover all concerns necessary to evaluate the overall system objective (Sage and Rouse 2009).

The first three evaluation measures were designed to measure the accuracy of feasibility assessments performed in Function 1.0, Assess Feasibility. Accuracy of the TRL was determined by the team to be critical in the success of the overall system. A failure to accurately identify the TRL jeopardizes the success of any program using the TDS. This measure will track how consistently the TRL was successfully identified at the beginning of a project. The accuracy of the technical and programmatic feasibility EMs were implemented to ensure the technology being developed is feasible - not a perpetual motion machine, for example - and the programmatic constraints of cost and schedule can be reasonably met by the TDS.

The second function, Function 2.0, Produce Technology Development Plan, was implemented to produce a plan, which identifies the risk, cost, and schedule for a technology in the TDS. The EMs assigned to measure the effectiveness of the plan are focused on identifying a high percentage of risks and producing accurate cost and schedule projections. An EM was also implemented to measure the percentage of plans that are both approved and signed by the customer upon first submittal. The EM was considered important in order to save critical schedule time by working with the customer up front and negotiate through issues while the report is written.

Function 3.0, Mature Technology, was assigned an EM for each of its top-level sub-functions. All of these EMs were designed to evaluate the requirements and specifications incorporated into the design and demonstrated by simulation and testing. Two of these focused on the design and build of prototypes. An EM was designated to ensure a high percentage of the design requirements were incorporated in the TDS prototype design and another EM defined that measures the actual number of specifications met when the prototype is built. The TDS specified prototypes should be simulated and tested in operational environments. These EMs were focused on a high percentage of performance requirements and operational performance criteria being demonstrated.

The Transition Technology function, Function 4.0, was developed to create customer documentation, validate the TRL through a TRA, and transition the project and all documentation to Function 6.0, Technology Maturation Closeout. The EM for finalizing the transition artifacts for the customer was based on the time required to finalize the product. This EM was intended to ensure when the technology is developed, it is then quickly documented and sent to the customer. The TRL validation EM serves the same purpose as described for Function 1.0. An EM was also included for measuring the number of final project artifacts that are delivered to the customer with no edits or reworks are required.

Function 5.0, Redefine/Terminate Program, was required to assess those projects that have not achieved the planned progress detailed in the Technology Development Plan, thus, being recommended for program plan redefinition or capturing the issues and terminating the program. The accuracy of the assessment will be critical to the success of DOD acquisition programs, so an EM was put in place to track the correctness of these assessments. As when the plan is first written, an EM was implemented to measure the percentage of plans that approved and signed by the customer at the first submittal. Since the reasons for terminating a project are critical and important to future DOD acquisition investments, an EM was put in place to measure the percentage of termination issues identified.

Two non-functional requirements, Repeatability and Usability, were also levied on the system and EMs were identified to measure their success. In order to ensure the TDS processes were repeatable, an EM was set up to measure the percentage of successful system executions. This EM defined a successful execution as projects that reach a TRL of 6 and are transitioned to the customer and projects in which critical issues are identified and a termination decision is made. To ensure the TDS is usable by all services within the DOD, the Usability requirement was implemented with the EM in place to measure the percentage of customers who were satisfied with the TDS.

These EMs provided metrics that were used in measuring the expected system performance of the TDS against the current prototyping process in the DOD. The EMs were necessary to ensure the objective of each function and requirement was met.

D. SUMMARY

The functional analysis included developing the functions, decomposition of the sub-functions and generation of IDEF0 diagrams for identification of the functional inputs and outputs. After functional decomposition and identification of system boundaries and functional inputs and outputs, requirements allocation was completed. Allocating system level requirements involved generating traceable system functional, input-output and non-functional requirements.

Analysis of the functions and system requirements drove the design of a value system for the TDS in order to develop metrics to evaluate the system. The team further developed the DOD prototyping improvement process idea with the generation of the value hierarchy. The value hierarchy's purpose to improve the DOD prototyping process, and for the outcome, the system needs satisfy

1) Assess the feasibility of technological development.
2) Be able to produce a complete and accurate technology development plan.
3) Mature the technology to a TRL 6.
4) Provide the activities necessary to transition the matured technology to the customer.
5) Provide the flexibility to redefine or terminate an ongoing technology development project should that need arise.
6) Be usable and repeatable to the intended users.

As the value hierarchy was delineated, design choices were identified to create potential concept solutions. However, these design choices could not feasibly satisfy all the top-level system requirements. An instantiated system solution could not be identified or documented. Because of this, the generic TDS is the only feasible solution to meet the top-level requirements. The TDS will only work in a synergistic manner with all functions relying on each other to improve the prototyping process. Completion of a value hierarchy provided the objectives and evaluation measures necessary to identify that the TDS is a suitable alternative to satisfy the need. Although a suitable instantiated alternative could not be found to satisfy the TDS concept, the team could not have known this until after completion of the value hierarchy, so its importance remained the same in the context of the systems engineering process.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. USE CASE AND SYSTEM MODEL

After the functional analysis and the value system design had been completed, the team was prepared to design the TDS model. The functional analysis, developed earlier in the SE process, formed the basis for the executable structure used to create the discrete event simulations. An executable architecture was developed to validate the system's functions and to evaluate the performance of the system. This architecture was a version of the system built within a modeling context that was capable of executing simulations that produced data for conducting analysis of the system (Pawlowski, Barr and Ring 2004). The TDS architecture was modeled with Innoslate, a systems engineering tool, developed by SPEC Innovations.

The purpose of the executable model was to create a unique discreet event simulation of the dynamic behavior of the TDS functions/processes (Buede 2009). In order to have a simulation context, use cases were identified and the resultant data used to test, verify, and validate the execution of the simulation environment. The use case data, derived from existing systems and program data, was provided as input into the system to evaluate the flow and logic of system functions and to evaluate the performance of the model. After executing the model, the team analyzed the output and results of the simulations.

A. SIMULATION MODEL

The purpose of the simulation model was to simulate the operation of the TDS functions. The model was developed as a proof of concept to emulate the architecture of the TDS and validate its operational merit. There were also necessary additions required by the simulation environment that will be discussed in the following sections. A series of use cases were selected as inputs both to validate the simulation structure itself, as well as to evaluate model performance as it relates to operational applicability.

Figure 38 displays the functional context diagram, which includes all of the top-level operational activities of the TDS. Figure 39 displays the TDS simulation model.

The following sections break down the model and describe each of the function and sub-functions.

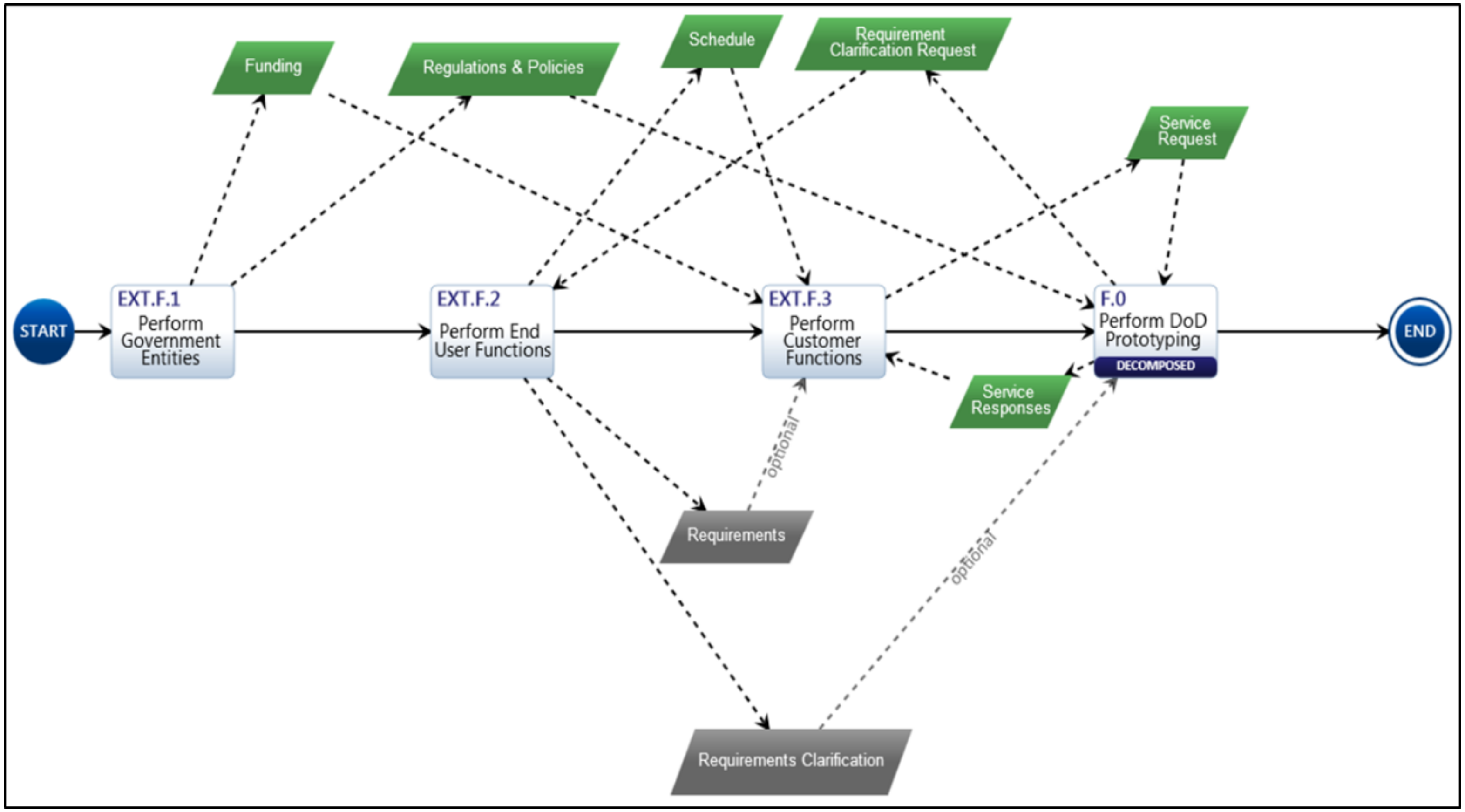


Figure 38. Functional Context Diagram

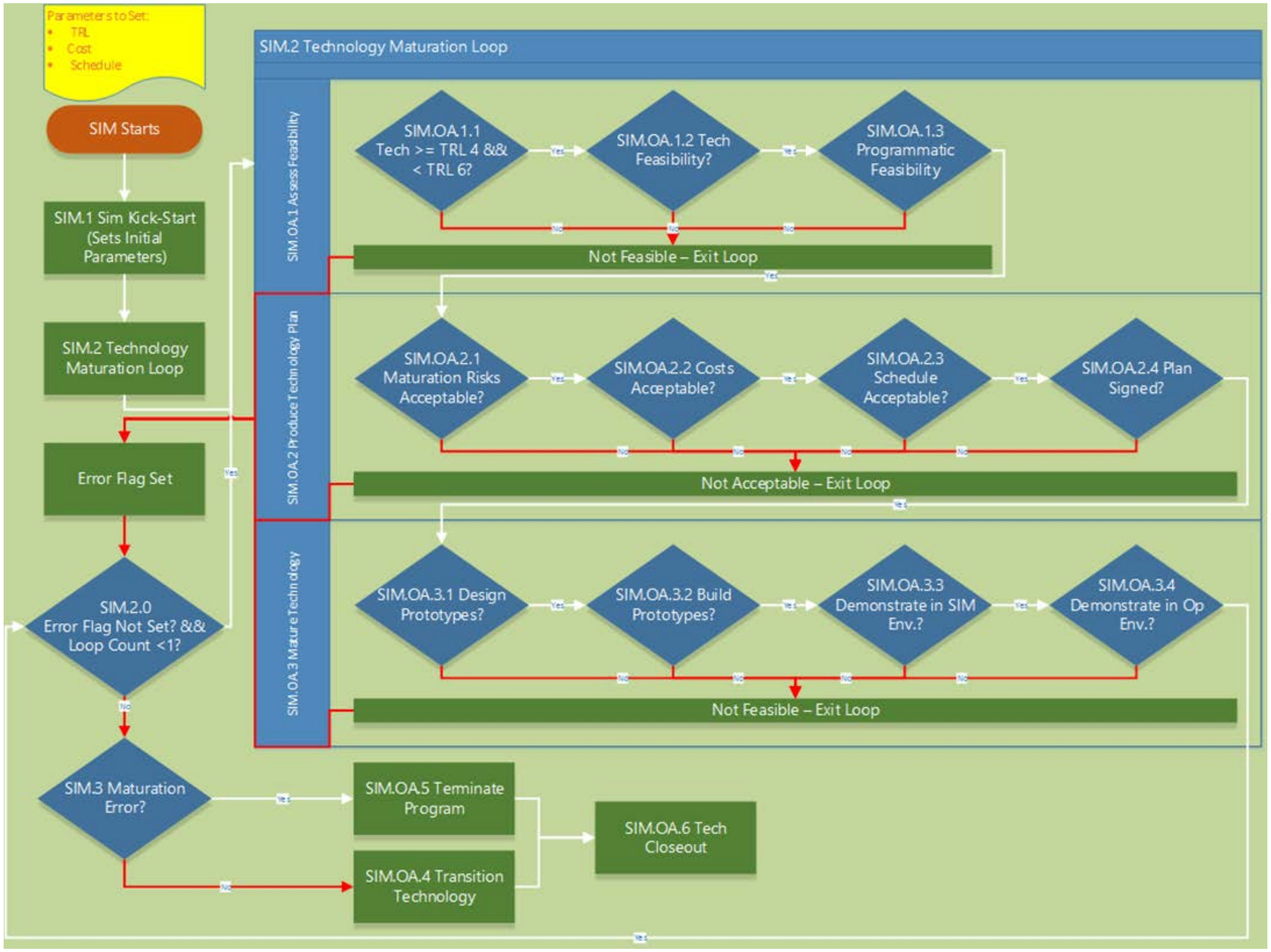


Figure 39. Simulation Functional Model

1. Simulation Specific Functions

The first set of functions were created for the simulation environment and not a part of the functionality of the TDS. The purpose of the simulation specific functions was to exert control over the simulation to ensure proper execution and to produce expected results. The simulation specific functions were required for proper control and interaction of the simulation.

a. SIM.1 – Sim Kick-Start

The Kick-Start function was established to set the global parameters provided from the use case. Figure 40 displays the Kick Start function modeled in Innoslate. The parameters that were to be set and utilized in the model were as follows:

- TRL: The TRL was provided from the use case and served as the earliest defined TRL assessment of the technology that was run through the simulation. Ideally the use case data would have provided a progression of the TRL maturing throughout the process. In order to meet system constraints and not generate an error, the selected TRL must be at a level of 4 or 5.
- Expected Schedule: The schedule for completion of technology development was provided from the use case and was to be captured as a snapshot in the technology development as possible. The Expected Schedule served as the baseline for program schedule slips during technology development.
- Expected Cost: The cost for completion of technology development was provided from the use case and was to be captured as a snapshot in the technology development as possible. The Expected Cost served as the baseline for program cost exceedances during technology development.
- Loop 1 Schedule: The schedule was captured from the first iteration of technology development, in order to define the “Actual” technology maturation schedule.
- Loop 2 Schedule: The schedule was captured from the second iteration of technology development, in order to define the “Actual” second iteration technology maturation schedule.

- Loop 1 Costs: The cost was captured from the first iteration of the technology development, in order to define the "Actual" technology maturation cost.
- Loop 2 Costs: The cost was captured from the second iteration of technology development, in order to define the "Actual" second iteration technology maturation cost.
- maturationCount: This variable served as a counter storing the number of times through the loop. Initialization of the maturationCount variable was set to "0."
- errorFlag: This variable was utilized to maintain the error state in the simulation. Initialization of the errorFlag variable was "False." The errorFlag would initialize should any fail condition occur during the simulation process.
- runningCosts: This variable was used for maintaining the running cost total between the loops.

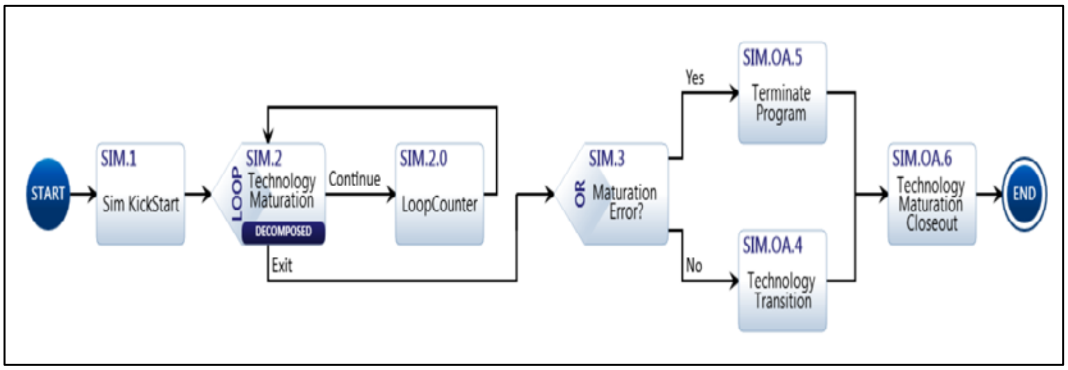


Figure 40. Sim Kick-Start System Model

b. SIM.2 – Technology Maturation Loop

The Technology Maturation Loop function enclosed the first three operational action functions inside a loop, since they had to be executed twice in a typical "ideal-path" scenario. For example, a technology that starts at TRL 4 would transition to TRL 5 during the first iteration, and subsequently, from TRL 5 to TRL 6 during the second iteration. This function was used to determine if the "errorFlag" had been set to "True." If errorFlag was set to "True," the simulation would exit the loop. If the "errorFlag" was set to "False," the function checks the "maturationCounter" value. If the "maturationCounter" value was equal to 1 or more, then the simulation exits the loop. If both of the two conditions are set to "False," then the loop continues to execute. The Technology Maturation Loop model is displayed in Figure 41.

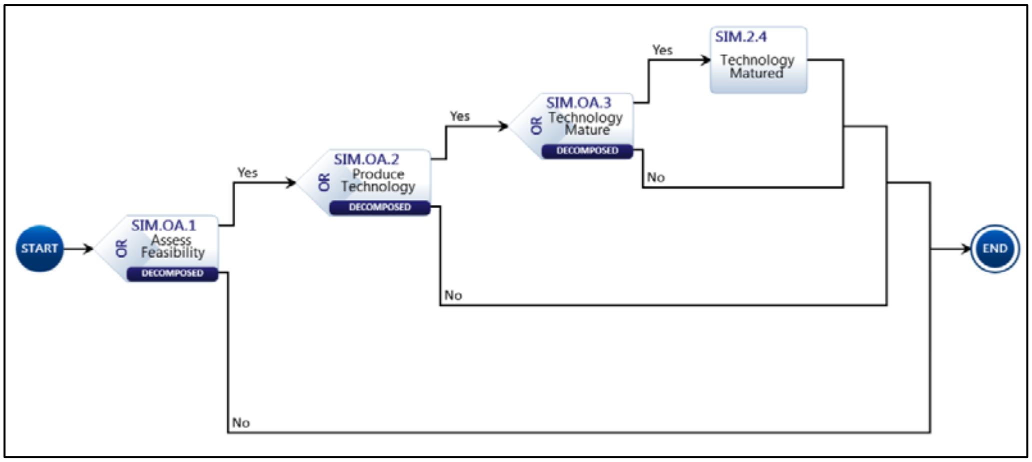


Figure 41. Decomposed Technology Maturation Loop Model

c. *SIM.2.0 – Loop Counter*

The Loop Counter was a function that incremented the "maturationCount" variable by a value of one. The Loop Counter changes with each individual iteration of the loop and was implemented to count the number of iterations of the loop.

d. *SIM.3 – Maturation Error?*

The purpose of the "maturationError?" function was to determine if the "errorFlag" had been set to "True." If the "errorFlag" was set to "True," the function was forced to continue to SIM.OA.5 Terminate Program. Otherwise, the "maturationError?" function continued to SIM.OA.4 Transition Technology.

2. Operational Action Functions

The operational action functions were developed to model the operation of the TDS by simulating each of the functions within the TDS. This section was meant to serve as a logic check of the architecture that had been developed. This set of simulation functions served to validate the simulation structure and model architecture.

a. SIM.OA.1 – Assess Feasibility

The SIM.OA.1, Assess Feasibility, function contained three sub-functions that determined if the use case technology progressed further into the model. This section explains the purpose and execution of those sub-functions. The Assess Feasibility function is shown in Figure 42.

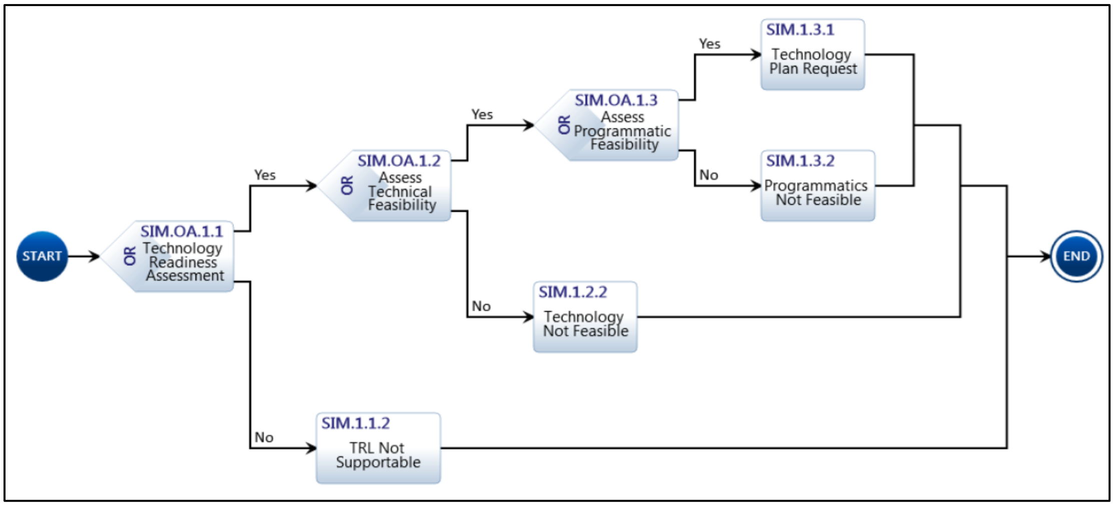


Figure 42. SIM.OA.1 Assess Feasibility Function

(1) SIM.OA.1.1 – Technology Readiness Assessment. The purpose of the SIM.OA.1.1, Technology Readiness Assessment, function was to ensure that the TRL of the use case technology being passed through this function was at a TRL of 4 or 5. If the TRL was not at 4 or 5, an error function was activated that set the “errorFlag” to “True” causing the simulation to exit the loop. Since this simulation was executed based on use case data, no actual TRA was performed nor was this function expected to change or redefine the TRL level that was input into the simulation. A value of “True” could be set manually allowing the simulation to pass through this SIM.OA.1.1 Technology Readiness Assessment function onto the next function.

(2) SIM.OA.1.2 – Technological Feasibility. The purpose of the SIM.OA.1.2, Technological Feasibility, function was to ensure the technology being requested for maturation was technically feasible. If it was determined infeasible to mature the technology, an error function was activated that set the “errorFlag” to “True” and resulted

in the simulation exiting the loop. Since this simulation was executed based on use case data, it was assumed that the maturation of the technology was feasible since it was pursued. This function was not expected to challenge nor disagree with the technical feasibility. A value of “True” could be set manually in order to allow the simulation to pass through this function.

(3) SIM.OA.1.3 – Programmatic Feasibility. The purpose of the SIM.OA.1.3, Programmatic Feasibility, function was to ensure that the programmatic aspects of the requested technology development and maturation effort was feasible. If the technology was found to be programmatically infeasible, an error function was activated that set the “errorFlag” to “True” and resulted in the simulation exiting the loop. Since this simulation was executed based on use case data, it was assumed that the maturation of the technology was feasible since it was pursued. This function was not expected to challenge nor disagree with the programmatic feasibility. A value of “True” could be set manually in order to allow the simulation to pass through this function.

b. SIM.OA.2 – Produce Technology Plan

The SIM.OA.2, Produce Technology Plan, function contained four sub-functions that determined if the use case technology progressed further into the model. This section explains the purpose and execution of those sub-functions. The Produce Technology Plan function is displayed in Figure 43.

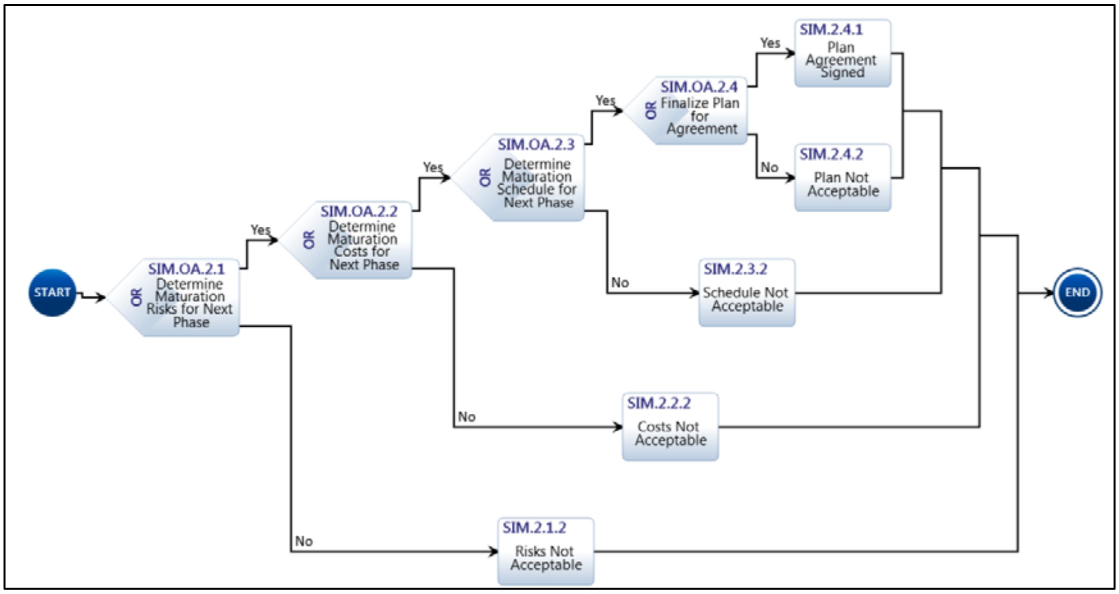


Figure 43. SIM.OA.2 Produce Technology Development Plan Function

(1) SIM.OA.2.1 – Maturation Risks for Next Phase. The purpose of the SIM.OA.2.1, Maturation Risks for Next Phase, function was to determine the maturation risk of the technology during the upcoming development phase. If the risks were found to be unacceptable by the customer, an error function was activated that set the “errorFlag” to “True” and resulted in the simulation exiting the loop. Since this simulation was executed based on use case data, it was assumed that the maturation risk of the technology was feasible since it was pursued. This function was not expected to create, change, or redefine risks. A value of “True” could be set manually in order to allow the simulation to pass through this function.

(2) SIM.OA.2.2 – Maturation Costs for Next Phase. The purpose of the SIM.OA.2.2, Maturation Costs for Next Phase, function was to determine the cost for maturating the technology during the upcoming development phase. If the cost was found to be unacceptable by the customer, an error function was activated that set the “errorFlag” to “True” and resulted in the simulation exiting the loop. Since this simulation was executed based on use case data, the function determined if the “Actual Cost” was more than the “Expected Cost.” If the “Actual Cost” was higher, it was deemed unacceptable and triggered an error. A value of “True” could be set manually in order to allow the simulation to pass through this function onto the next function.

(3) SIM.OA.2.3 – Maturation Schedule for Next Phase. The purpose of the SIM.OA.2.3, Maturation Schedule for Next Phase, function was to determine the technology maturation schedule during the upcoming development phase. If the schedule was found unacceptable by the customer, an error function was activated that set the "errorFlag" to "True" and resulted in the simulation exiting the loop. This simulation is based on use case data, so SIM.OA.2.3 will determine if the "Actual Schedule," derived from the use case, exceeds the "Expected Schedule" for technology maturation. If so, the costs will be deemed unacceptable and will trigger an error. A value of "True" could be set manually in order to allow the simulation to pass through this function.

(4) SIM.OA.2.4 – Maturation Plan Agreement Signed. The purpose of the SIM.OA.2.4, Maturation Plan Agreement Signed, function was to ensure the maturation plan agreement was approved and signed by the customer. If the plan was deemed unacceptable, an error function was activated setting the "errorFlag" to "True" and resulted in the simulation exiting the loop. A value of "True" could be set manually in order to allow the simulation to pass through this function onto the next function.

c. SIM.OA.3 – Mature Technology

The SIM.OA.3, Mature Technology, function contained four sub-functions. These sub-functions determined if the use case technology progressed further into the model. This section explains the purpose of those sub-functions.

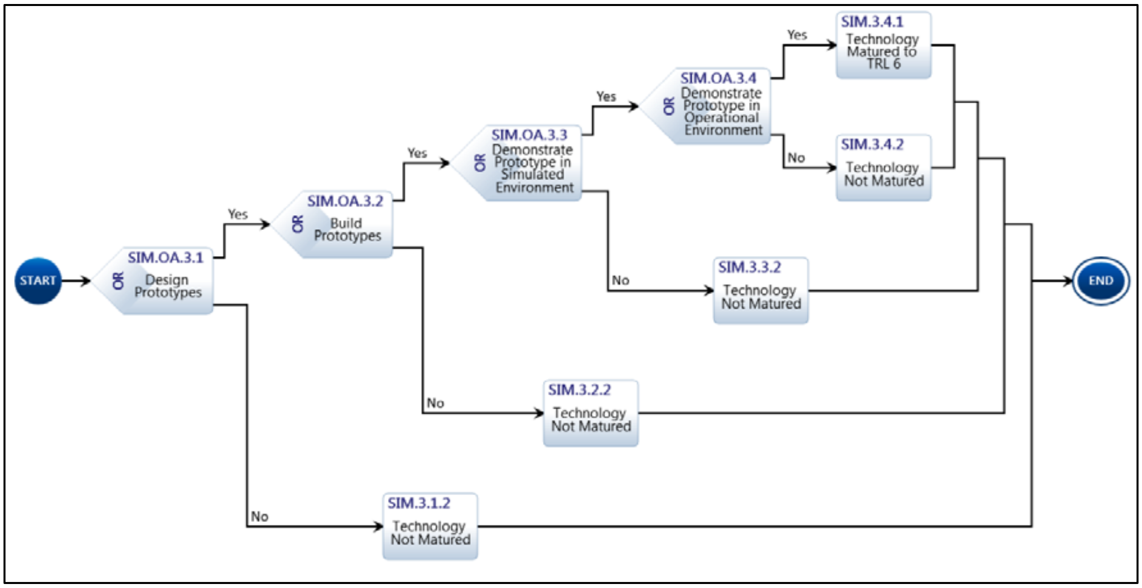


Figure 44. SIM.OA.3 Mature Technology Function

(1) SIM.OA.3.1 – Design Prototype. The SIM.OA.3.1, Design Prototype, modeled the design of the prototype. While the SIM.OA.3.1 could not re-enact the Design Prototype process that was defined for the system, it was able to serve as placeholders for the process. SIM.OA.3.1 also had a time variable associated with the function that could have been applied if required.

(2) SIM.OA.3.2 – Build Prototype. The SIM.OA.3.2, Build Prototype, modeled the building of the prototype. While the SIM.OA.3.2 could not re-enact the Build Prototype process that was defined for the system, it was able to serve as placeholders for the process. SIM.OA.3.2 also had a time variable associated with the function that could have been applied if required.

(3) SIM.OA.3.3 – Demonstrate Prototype in a Simulated Environment. The SIM.OA.3.3, Demonstrate Prototype in a Simulated Environment, modeled the prototype demonstrating a simulated design reference mission in order to validate operation in a simulated operational environment. While the SIM.OA.3.3 could not re-enact the Demonstrate Prototype in a simulated environment process defined for the system, it was able to serve as placeholders for the process. Also SIM.OA.3.3 had a time variable associated with the function that could have been applied if required.

SIM.OA.3.4 – Demonstrate Prototype in an Operational Environment. The purpose of the SIM.OA.3.4 Demonstrate Prototype in an Operational Environment function was to model the demonstration of the prototype in a relative user environment. While the SIM.OA.3.4 could not re-enact the Demonstrate Prototype in an operational environment process that was defined for the system, it was able to serve as placeholders for the process. Also SIM.OA.3.4 had a time variable associated with the function that could have been applied if required.

d. *SIM.OA.4 – Transition Technology*

The purpose of the SIM.OA.4 Transition Technology function was to model the activity of passing the technology that was developed and matured to the customer. While the SIM.OA.4 could not re-enact the Transition Technology process that was defined for the system, it was able to serve as placeholders for the process. Also SIM.OA.4 had a time variable associated with the function that could have been applied if required.

e. *SIM.OA.5 – Terminate Program*

The purpose of the SIM.OA.5 Terminate Program function was to model the activity of terminating the technology development effort based on any number factors. While the SIM.OA.5 could not re-enact the Terminate Program process that was defined for the system, it was able to serve as placeholders for the process. Also SIM.OA.5 had a time variable associated with the function that could have been applied if required.

f. *SIM.OA.6 – Technology Program Closeout*

The purpose of the SIM.OA.6 Technology Program Closeout function was to model the activity of closing out the technology development effort whether through technology transition or termination. While the SIM.OA.6 could not re-enact the Technology Program Closeout process that was defined for the system, it was able to serve as placeholders for the process. Also SIM.OA.6 had a time variable associated with the function that could have been applied if required.

B. USE CASES

The team developed use cases to evaluate the performance of the system simulation model. Each of the eight use cases provide relevant examples that could occur during system operation. In this section, the team discusses the parameters, expected results, and actual results of each use case. The simulations conducted in Innoslate provide a Gant chart output illustrating the simulation timeline. Explanation of the simulation details described in the Gant charts is provided in Appendix E.

1. Use Case 1 – Ideal Path

The Ideal Path use case represented an example with the ideal parameters. Table 11 displays the simulation model input parameters. This use case served as a simulation verification test to verify that the Ideal Path was executable, given the proper parameters.

Table 11. Ideal Path Use Case Input Parameters

Parameter	Value
TRL	‘4’
Total Expected Schedule	‘2016’
Loop 1 Schedule	‘2015’
Loop 2 Schedule	‘2016’
Total Expected Costs	‘1000000’ ($1,000,000)
Loop 1 Costs	‘500000’ ($500,000)
Loop 2 Costs	‘500000’ ($500,000)

The expected execution of the simulation was to run through the maturation loop twice, to pass to the Transition Technology function, and to complete the simulation with the Maturation Closeout function. The simulation in Innoslate provides a Gant Chart, as shown in Figure 45, illustrating the actual result of the simulation on a timeline describing when each function was utilized. The timeline for the functions is arbitrarily set and not indicative of actual function duration. The results, as expected, demonstrated that the technology matured through two loops of the simulation with no errors until technology maturity reached a TRL 6 and successfully transitioned to the next phase of the life cycle. This indicated that the simulation was built such that, when given the required inputs, parameters, and expected values, the results are successful. Figure 45 displays the actual results of the simulation with the ideal path.

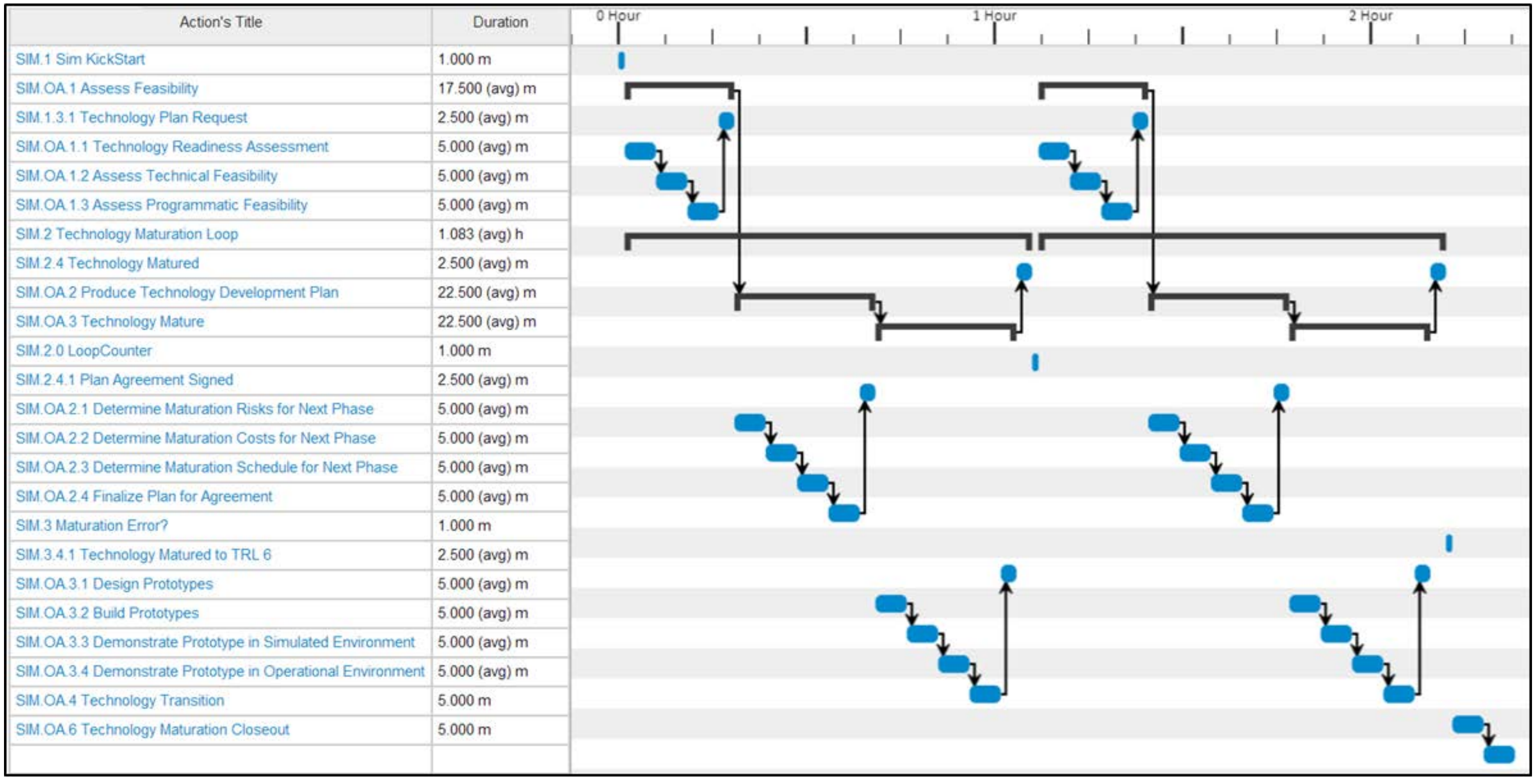


Figure 45. Ideal Path Use Case Simulation Results

2. Use Case 2 – Technology Readiness Assessment (TRA) Error out TRL Less than 4

The “TRA Error out TRL Less than 4” use case represents an example of a technology entering the system simulation at a TRL less than 4. This use case served as a simulation verification test to verify that the “error out TRL less than 4” path was executable, given the proper parameters. Table 12 displays the TRA Error out TRL less than four simulation model input parameters.

Table 12. TRA Error out TRL Less than 4 Use Case Input Parameters

Parameter	Value
TRL	**‘3’**
Total Expected Schedule	‘2016’
Loop 1 Schedule	‘2015’
Loop 2 Schedule	‘2016’
Total Expected Costs	‘1000000’ ($1,000,000)
Loop 1 Costs	‘500000’ ($500,000)
Loop 2 Costs	‘500000’ ($500,000)

The expected execution of the simulation was to enter the maturation loop and trigger an error in function SIM.OA.1.1 TRA After the error message was generated, the simulation would exit the loop and pass the message to the Terminate Program function. To complete the simulation process, the Terminate Program function passed the error message to the Maturation Closeout function.

Figure 46 displays the results of the simulation when the technology TRL of less than 4 was processed by the model. The simulation completed SIM.OA.1.1 Technology Readiness Assessment and proceeded to SIM 1.1.2 TRL Not Supportable, indicating that the technology was not equal to TRL 4 or 5. Next, the logic gate SIM.3 “Maturation Error?” was processed and the simulation continued to the SIM OA.5 Terminate Program and SIM OA.6 Technology Maturation Closeout functions. This was the logical path for the TDS system to take if a TRL 3 technology entered the system. This outcome confirmed that the model was properly checking the lower bound of the TRL upon entry into the system.

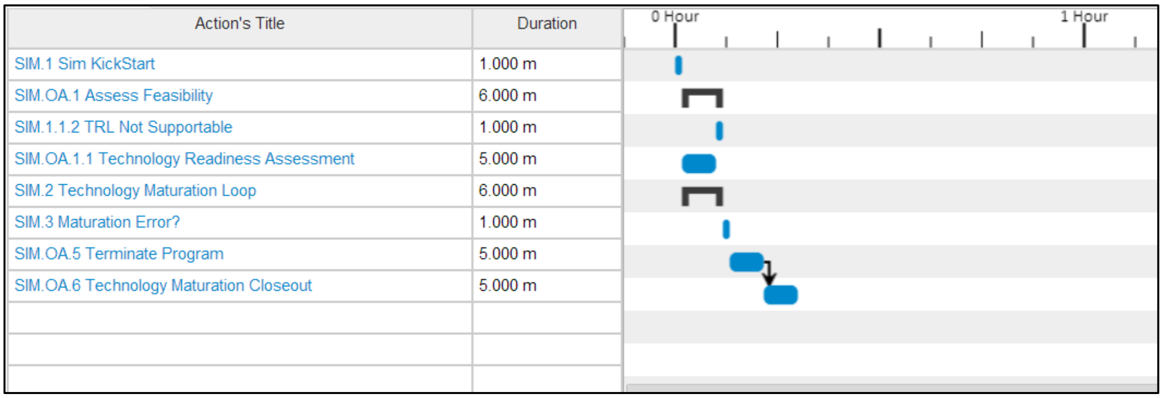


Figure 46. Use Case 2 Simulation Results

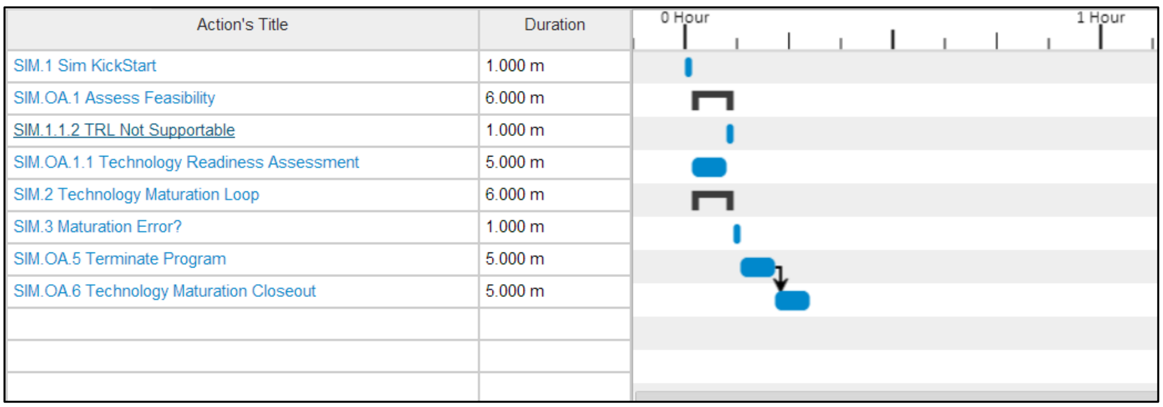
3. Use Case 3 – TRA Error out TRL Greater than 5

The "TRA Error out TRL Greater than 5" use case provided an example of a technology entering the system simulation at a TRL greater than 5. This use case served as a simulation test to verify that the error out TRL greater than 5 path was executable, given the proper parameters. Table 13 displays the "TRA Error out TRL greater than 5" simulation model input parameters.

Table 13. TRA Error out TRL Greater than 5 Use Case Input Parameters

Parameter	Value
TRL	**'6'**
Total Expected Schedule	'2016'
Loop 1 Schedule	'2015'
Loop 2 Schedule	'2016'
Total Expected Costs	'1000000' ($1,000,000)
Loop 1 Costs	'500000' ($500,000)
Loop 2 Costs	'500000' ($500,000)

The expected execution of the simulation was to enter into the maturation loop and trigger an error in function SIM.OA.1.1 TRA. After the error message was generated, the simulation would exit the loop and pass the message to the Terminate Program function. To complete the simulation process, the Terminate Program function passed the error message to the Maturation Closeout function. The results of the simulation mirrored the expected outcome as shown in Figure 47. The system correctly produced an error when the technology was matured beyond TRL 5.



Action's Title	Duration
SIM.1 Sim KickStart	1.000 m
SIM.OA.1 Assess Feasibility	6.000 m
SIM.1.1.2 TRL Not Supportable	1.000 m
SIM.OA.1.1 Technology Readiness Assessment	5.000 m
SIM.2 Technology Maturation Loop	6.000 m
SIM.3 Maturation Error?	1.000 m
SIM.OA.5 Terminate Program	5.000 m
SIM.OA.6 Technology Maturation Closeout	5.000 m

Figure 47. Use Case 3 Simulation Results

4. Use Case 4 – Expected Costs Error Out in Loop 1

The “Expected Costs Error Out in Loop 1” use case provided an example in which the Loop 1 cost exceeded the total expected cost creating a simulation error. This use case served as a simulation test to verify that the expected cost error out in Loop 1 path was executable, given the proper parameters. Table 14 displays the expected cost error out for the Loop 1 simulation model input parameters.

Table 14. Expected Costs Error Out in Loop 1 Use Case Input Parameters

Parameter	Value
TRL	‘4’
Total Expected Schedule	‘2016’
Loop 1 Schedule	‘2015’
Loop 2 Schedule	‘2016’
Total Expected Costs	**‘100000’ ($100,000)**
Loop 1 Costs	**‘500000’ ($500,000)**
Loop 2 Costs	‘500000’ ($500,000)

The expected execution of the simulation was to enter into the maturation loop and trigger an error in function SIM.OA.2.2 Maturation Costs for Next Phase. After the error message was generated, the simulation would exit the loop and pass the message to the Terminate Program function. To complete the simulation process, the Terminate Program function passed the error message to the Maturation Closeout function. The simulation results are shown in Figure 48, which accurately present the system model reaction when the Loop 1 cost exceeded the Total Expected Cost.

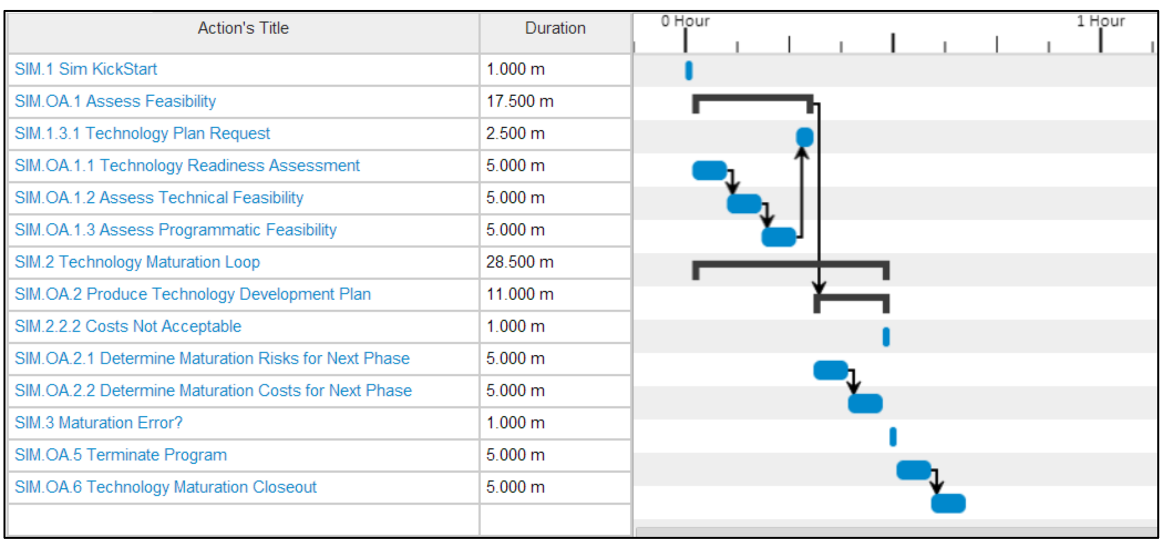


Figure 48. Use Case 4 Simulation Results

5. Use Case 5 – Expected Costs Error Out in Loop 2

The "Expected Costs Error Out in Loop 2" use case provided an example in which the Loop 2 cost did not align with the total expected cost creating a simulation error. This use case served as a simulation test to verify that the expected cost error out in Loop 2 path was executable, given the proper parameters. Table 15 displays the expected cost error out for the Loop 2 simulation model input parameters.

Table 15. Expected Costs Error Out in Loop 2 Use Case Input Parameters

Parameter	Value
TRL	'4'
Total Expected Schedule	'2016'
Loop 1 Schedule	'2015'
Loop 2 Schedule	'2016'
Total Expected Costs	**'100000' ($100,000)**
Loop 1 Costs	'50000' ($50,000)
Loop 2 Costs	**'500000' ($500,000)**

The expected execution of the simulation was to enter into the maturation loop, pass through the loop once, and trigger an error in function SIM.OA.2.2 Maturation Costs for Next Phase. After the error message was generated, the simulation would exit the loop and pass the message to the Terminate Program function. To complete the simulation process, the Terminate Program function passed the error message to the Maturation Closeout function. The results from the simulation, shown in Figure 49, show that the simulation properly generated an error during the second iteration of technology development when the cost exceeded the Total Expected Cost.

Action's Title	Duration
SIM.1 Sim KickStart	1.000 m
SIM.OA.1 Assess Feasibility	17.500 (avg) m
SIM.1.3.1 Technology Plan Request	2.500 (avg) m
SIM.OA.1.1 Technology Readiness Assessment	5.000 (avg) m
SIM.OA.1.2 Assess Technical Feasibility	5.000 (avg) m
SIM.OA.1.3 Assess Programmatic Feasibility	5.000 (avg) m
SIM.2 Technology Maturation Loop	14.792 (avg) h
SIM.2.4 Technology Matured	2.500 m
SIM.OA.2 Produce Technology Development Plan	16.750 (avg) m
SIM.OA.3 Technology Mature	22.500 m
SIM.2.0 LoopCounter	1.000 m
SIM.2.2.2 Costs Not Acceptable	1.000 m
SIM.2.4.1 Plan Agreement Signed	2.500 m
SIM.OA.2.1 Determine Maturation Risks for Next Phase	5.000 (avg) m
SIM.OA.2.2 Determine Maturation Costs for Next Phase	5.000 (avg) m
SIM.OA.2.3 Determine Maturation Schedule for Next Phase	5.000 m
SIM.OA.2.4 Finalize Plan for Agreement	5.000 m
SIM.3 Maturation Error?	1.000 m
SIM.3.4.1 Technology Matured to TRL 6	2.500 m
SIM.OA.3.1 Design Prototypes	5.000 m
SIM.OA.3.2 Build Prototypes	5.000 m
SIM.OA.3.3 Demonstrate Prototype in Simulated Environment	5.000 m
SIM.OA.3.4 Demonstrate Prototype in Operational Environment	5.000 m
SIM.OA.5 Terminate Program	5.000 m
SIM.OA.6 Technology Maturation Closeout	5.000 m

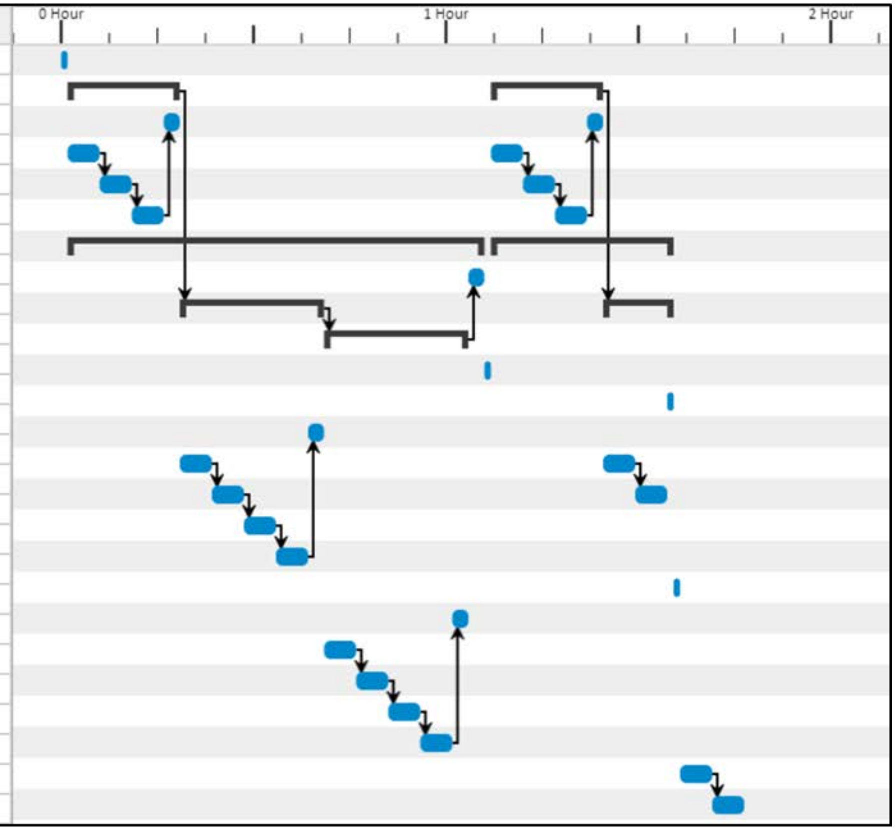


Figure 49. Use Case 5 Simulation Results

6. Use Case 6 – Schedule Error Out Loop 1

The “Schedule Error Out in Loop 1” use case provided an example in which the Loop 1 schedule did not align with the total expected schedule creating a simulation error. This use case served as a simulation test to verify that the schedule error out in Loop 1 path was executable, given the proper parameters. Table 16 displays the schedule error out for the Loop 1 simulation model input parameters.

Table 16. Schedule Error Out Loop 1 Use Case Input Parameters

Parameter	Value
TRL	‘4’
Total Expected Schedule	**‘2014’**
Loop 1 Schedule	**‘2015’**
Loop 2 Schedule	‘2016’
Total Expected Costs	‘1000000’ ($1,000,000)
Loop 1 Costs	‘500000’ ($500,000)
Loop 2 Costs	‘500000’ ($500,000)

The expected execution of the simulation was to enter into the maturation loop and trigger an error in function SIM.OA.2.3 Maturation Schedule for Next Phase. After the error message was generated, the simulation would exit the loop and pass the message to the Terminate Program function. To complete the simulation process, the Terminate Program function passed the error message to the Maturation Closeout function. The results of the simulation are shown in Figure 50, which properly recognized that the schedule during its first iteration exceeded the total schedule. As a result, the simulation generated an error and terminated the program.

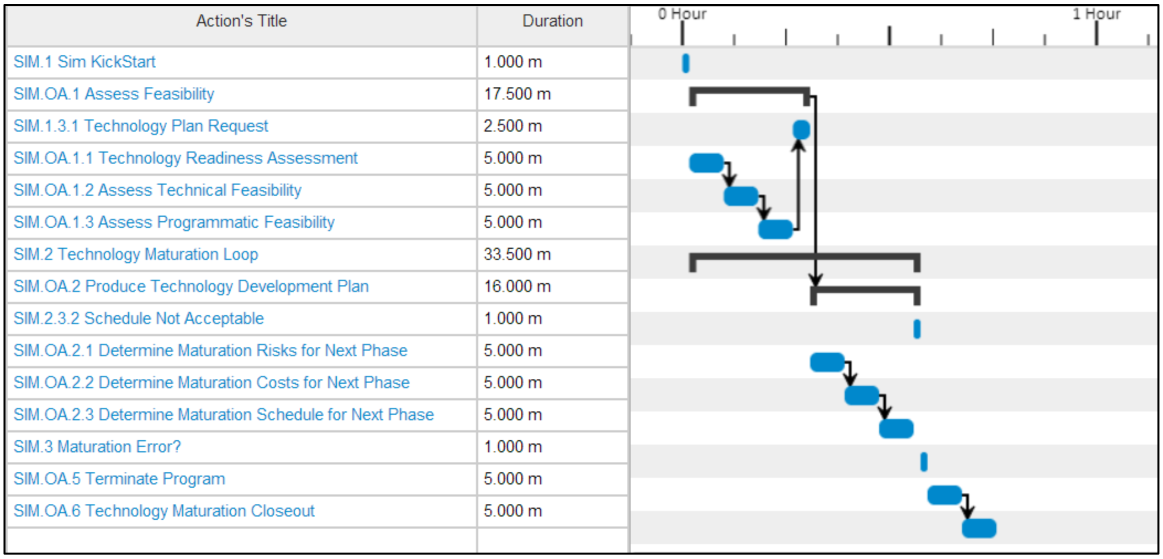


Figure 50. Use Case 6 Simulation Results

7. Use Case 7 – Schedule Error Out Loop 2

The "Schedule Error Out in Loop 2" use case provided an example in which the Loop 2 schedule did not align with the total expected schedule creating the simulation error. This use case served as a simulation test to verify that the schedule error out in Loop 2 path was executable, given the proper parameters. Table 17 displays the schedule error out for the Loop 2 simulation model input parameters.

Table 17. Schedule Error Out Loop 2 Use Case Input Parameters

Parameter	**Value**
TRL	'4'
Total Expected Schedule	**'2015'**
Loop 1 Schedule	'2015'
Loop 2 Schedule	**'2016'**
Total Expected Costs	'1000000' ($1,000,000)
Loop 1 Costs	'500000' ($500,000)
Loop 2 Costs	'500000' ($500,000)

The expected execution of the simulation was to enter into the maturation loop, pass through the loop once and trigger an error in function SIM.OA.2.3 Maturation Schedule for Next Phase. After the error message was generated, the simulation would exit the loop and pass the message to the Terminate Program function. To complete the simulation process, the Terminate Program function passed the error message to the Maturation Closeout function. The result of the simulation, in Figure 51, shows that the system model responded properly with the expected schedule exceeding the Total Expected Schedule. As expected, the model generated an error and terminated the program.

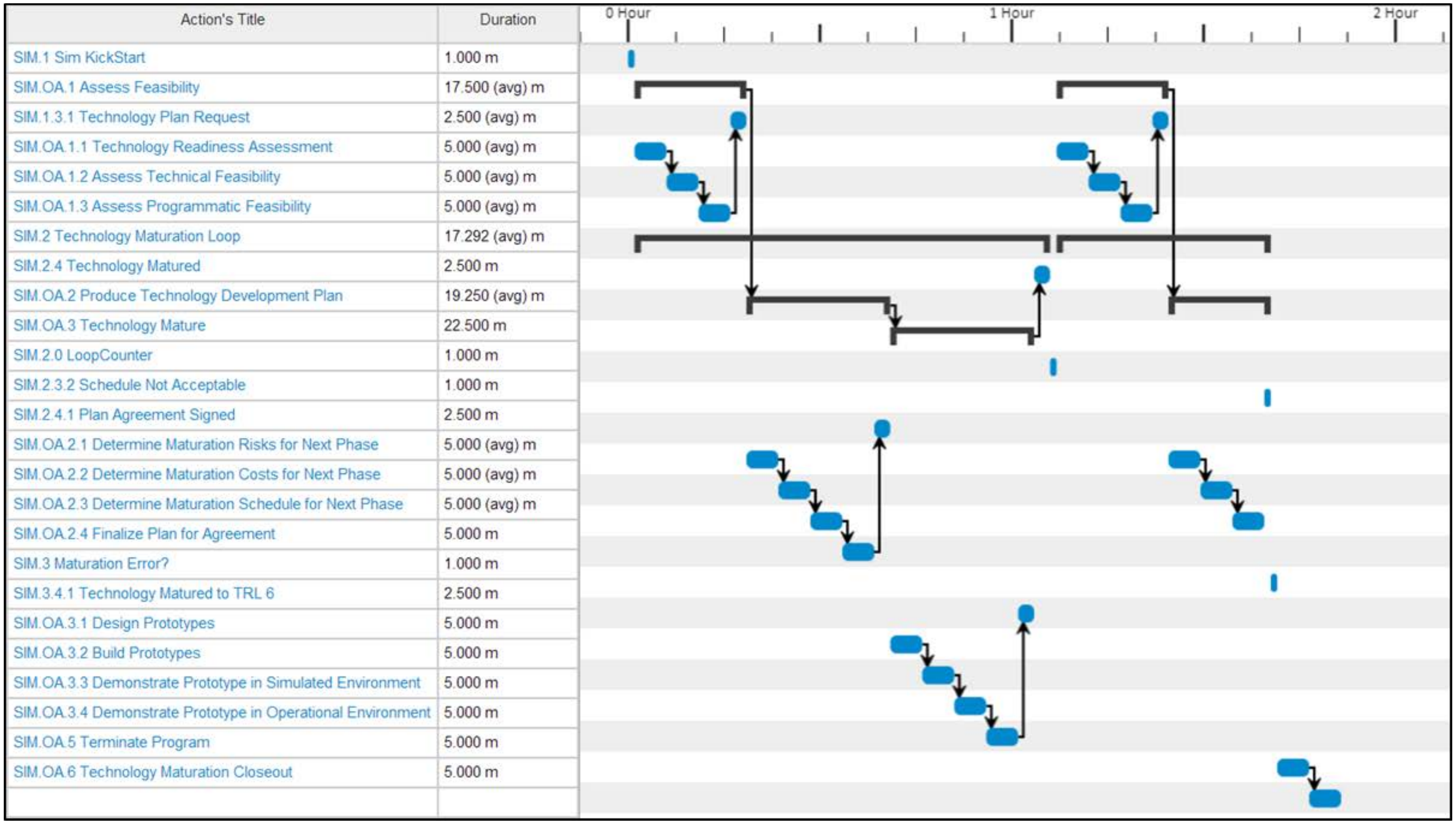


Figure 51. Use Case 7 Simulation Results

8. The Weapons System Use Cases

The purpose of the weapon system use cases was to provide the TDS with real world data from actual weapon system development programs in order to verify that TDS would be capable of producing similar or improved results above that of the final program baselines. The specific weapon system programs that were selected to provide input parameters for the system model were the Future Combat System (FCS) and the Apache Manned/Unmanned Common Architecture (MCAP).

a. Use Case 8 – FCS Case Study

The FCS was a multibillion dollar Army program that was intended to replace the current structure of force with modular systems in a common network structure. Its development utilized a system of systems (SOS) approach consisting of eighteen manned and unmanned systems to be integrated together by an extensive communications and information network (GAO 2008).

The FCS program began in May 2003 and was expected to be completed in three years with an estimated cost of $18 billion. In 2006, its schedule was extended to 2009, and its total cost grew to $25 billion (GAO 2008). The comparison of the original cost estimate and the extended cost is displayed in Table 18. In order to input FCS into the TDS simulation model, the team extracted the specific parameter values required by the TDS simulation model. The input parameters that were used for the FCS Use Case are shown in Table 19. Figure 52 depicts the simulation global variables that are scripted as part of the Sim Kick-Start function. Upon completion of the simulation scripts, the team executed the model and the results of the simulation are shown in Figure 53.

Table 18. FCS Cost Comparison (from GAO 2008)

Table 3: Comparison of the Original Cost Estimate and Recent Cost Estimates for the FCS Program (in billions of dollars)

	May 2003 Army estimate	December 2005 Army estimate	May 2006 CAIG estimate	December 2006 Army estimate	April 2007 IDA assessment
Base-year 2003 Dollars					
Research, development, testing, and evaluation	$18.1	$26.4	$31.8 - 44.0	$25.1	Approx $38.1
Procurement	$59.1	$92.8	$118.7	$87.5	N/A
Total	**$77.2**	**$119.2**	**$150.5 - 162.7**	**$112.6**	**N/A**
Then-year Dollars					
Research, development, testing, and evaluation	$19.6	$30.6	$36.6 - 52.7	$29.3	N/A
Procurement	$71.8	$133.1	$166.7 - 181.2	$131.6	N/A
Total	**$91.4**	**$163.7**	**$203.3 - 233.9**	**$160.9**	**N/A**

Source: U.S. Army, Office of the Secretary of Defense, IDA (data); GAO (analysis and presentation).

Table 19. FCS Cost Comparison (after GAO 2008)

Parameter	Value
TRL	'4'
Total Expected Schedule	'2006'
Loop 1 Schedule	'2006'
Loop 2 Schedule	'2009'
Total Expected Costs	'18000000000' ($18 billion)
Loop 1 Costs	'18000000000' ($18 billion)
Loop 2 Costs	'25000000000' ($25 billion)

In order to prepare the TDS model for the FCS simulation, the team developed specific scripts within the Innoslate modeling tool. Figure 52 depicts a sample script, named Sim Kick-Start that was developed to begin the execution of the FCS simulation in Innoslate. Sim Globals, a sub-function within the script, sets the input parameters to default values. The values utilized for these parameters were based on information collected from FCS documentation.

Based on FCS data, its initial TRL value was set to four and the other parameter default values were set zero (GAO 2006). The System Expected Values for the schedule and cost value parameters were set as shown in the Figure 52, as well as, values the for

Loop Schedule and Loop Costs parameters. Upon completion of the simulation scripts, the team executed the model. The results of the simulation are shown in Figure 53.

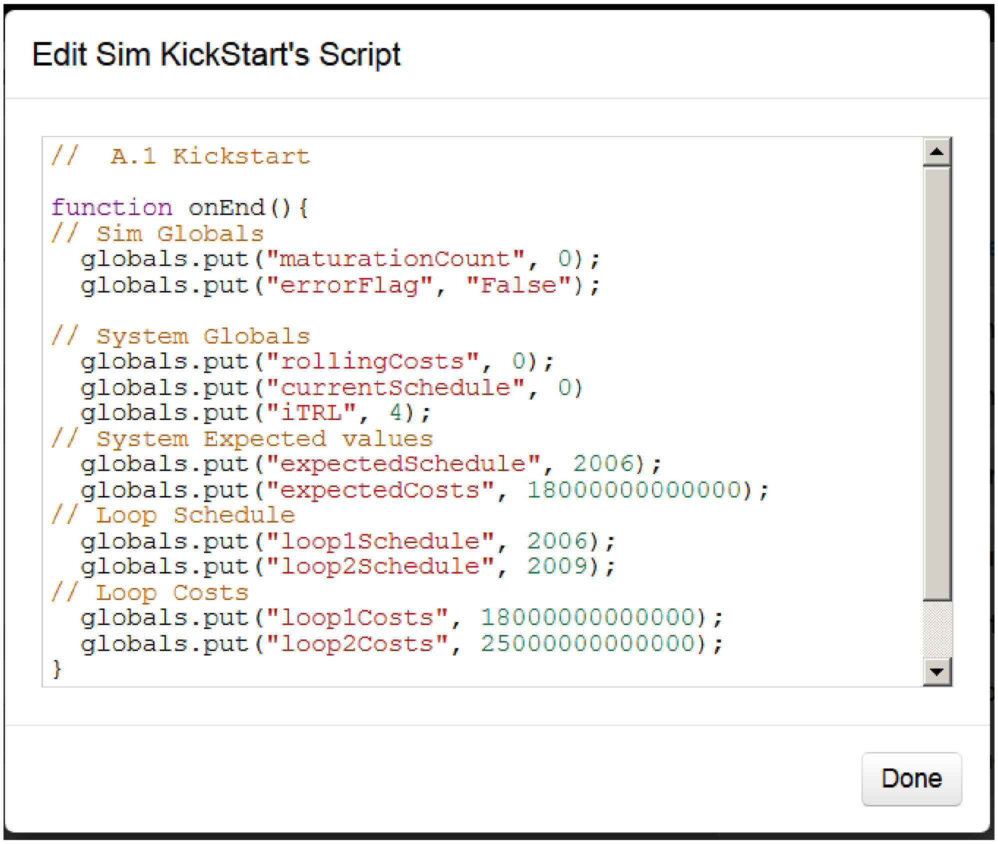
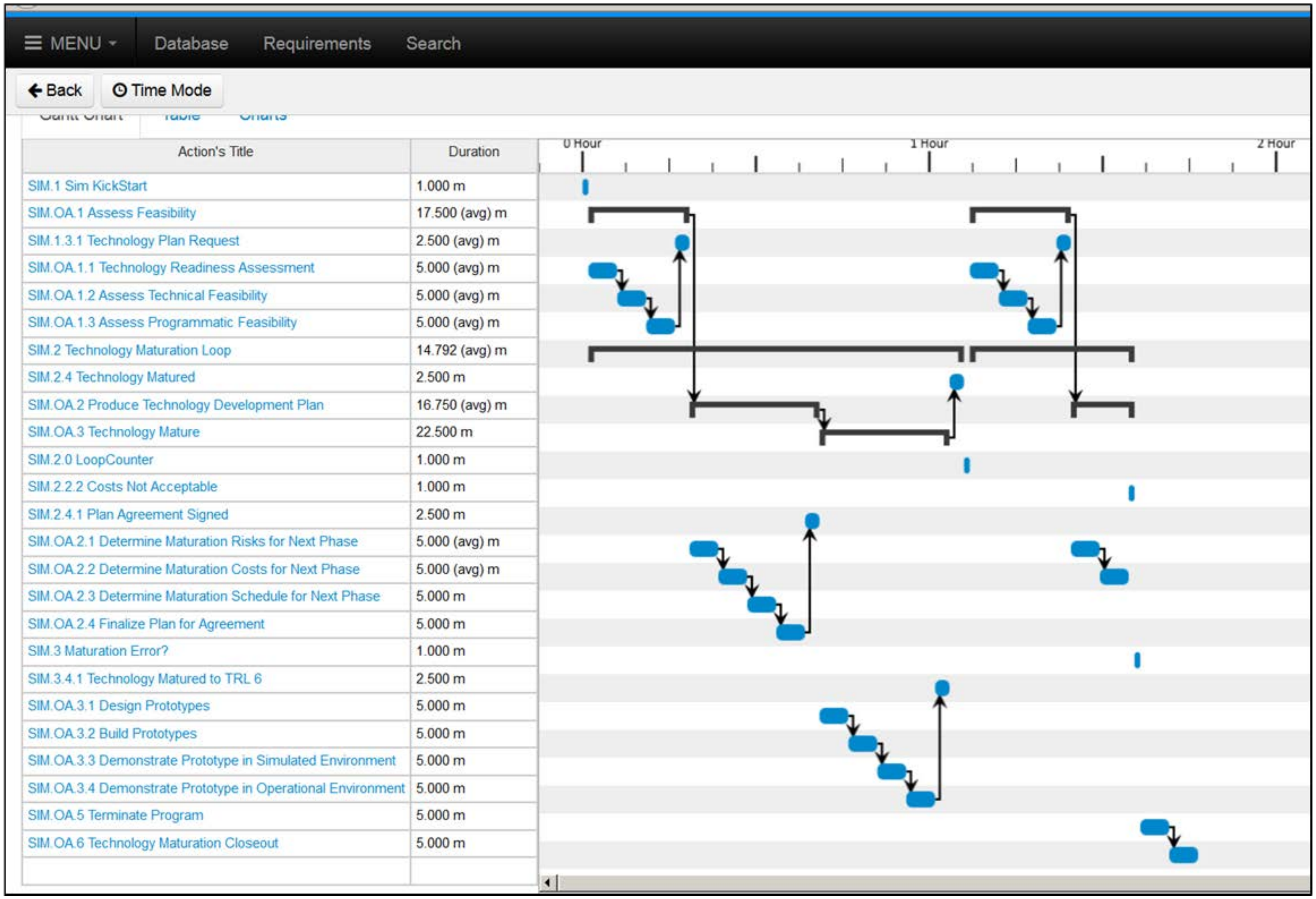


Figure 52. FCS Use Case Model Scripts



Action's Title	Duration
SIM.1 Sim KickStart	1.000 m
SIM.OA.1 Assess Feasibility	17.500 (avg) m
SIM.1.3.1 Technology Plan Request	2.500 (avg) m
SIM.OA.1.1 Technology Readiness Assessment	5.000 (avg) m
SIM.OA.1.2 Assess Technical Feasibility	5.000 (avg) m
SIM.OA.1.3 Assess Programmatic Feasibility	5.000 (avg) m
SIM.2 Technology Maturation Loop	14.792 (avg) m
SIM.2.4 Technology Matured	2.500 m
SIM.OA.2 Produce Technology Development Plan	16.750 (avg) m
SIM.OA.3 Technology Mature	22.500 m
SIM.2.0 LoopCounter	1.000 m
SIM.2.2.2 Costs Not Acceptable	1.000 m
SIM.2.4.1 Plan Agreement Signed	2.500 m
SIM.OA.2.1 Determine Maturation Risks for Next Phase	5.000 (avg) m
SIM.OA.2.2 Determine Maturation Costs for Next Phase	5.000 (avg) m
SIM.OA.2.3 Determine Maturation Schedule for Next Phase	5.000 m
SIM.OA.2.4 Finalize Plan for Agreement	5.000 m
SIM.3 Maturation Error?	1.000 m
SIM.3.4.1 Technology Matured to TRL 6	2.500 m
SIM.OA.3.1 Design Prototypes	5.000 m
SIM.OA.3.2 Build Prototypes	5.000 m
SIM.OA.3.3 Demonstrate Prototype in Simulated Environment	5.000 m
SIM.OA.3.4 Demonstrate Prototype in Operational Environment	5.000 m
SIM.OA.5 Terminate Program	5.000 m
SIM.OA.6 Technology Maturation Closeout	5.000 m

Figure 53. FCS Use Case Simulation Results

The simulation execution was expected to enter into the maturation loop and pass through the loop once. Upon entering the maturation loop the second time an error was expected to be triggered in function SIM.OA.2.2 Maturation Costs for Next Phase. The reason that this error was expected is because the total expected costs of $18 billion was expended in the first loop so there would have been no more funding available to proceed through the second loop. The error message triggered the simulation to exit out of the loop and pass the message to the Terminate Program function. The Terminate Program function recognizing that the error flag had been set would send the simulation to the Terminate Program function and then finished up with the Maturation Closeout function.

It is important to highlight that one of the constraints of the simulation is that the simulation can only provide a limited analysis based off of the quality of the data that is provided into the model. The FCS data was gathered from available online searches and was not to the level of detail and granularity that would have been preferred. Therefore, it is impossible to state unequivocally, that the TDS system approach would have resulted in a successful development of the FCS system. However, the resulting data shows that had the TDS system approach been used and resulted in similar results as the actual development, the TDS system would have raised issues about the technology's lack of mature ability much earlier in the process.

The original schedule showed that FCS was expected to have its technologies at a TRL of 6 by 2006 and within a budget of $18 billion but the actual development that took place up to 2006 was only able to advance the technologies from a TRL of 4 to a TRL of 5 at a cost of $18 billion. The TDS system has several instances throughout the process, that, had it been utilized, would of highlighted the issue much sooner than 2006. As an example during the planning phase of the TDS system, if the system had come back with a detailed cost and schedule that would have shown the development only reaching a TRL of 5 after having expended the entire budget and used the entire schedule, this would have raised the first red flag. If the system had projected that TRL 5 would have been reached in 2004 yet the system was still going through maturation and had not matured to the level expected and as expected, this also would have raised a red flag.

Based on the data that was available and the current data checks that had been incorporated into the TDS simulation, the FCS use case simulation ran through the initial maturation loop and encountered an error during the execution of the loop 2 costs function because the combined total of the cost for loop 1 and loop 2 exceeded the total expected costs amount. This resulted in the simulation running the project termination function and finishing with the program closeout function. This behavior resulted as expected.

b. Use Case 9 – Apache MCAP Case Study

The objective of the Apache MCAP was to develop and demonstrate an affordable high-performance embedded mission avionics processing architecture that could be utilized by manned rotorcraft platforms. Its development was based on open systems architecture standards and commercial-off-the-shelf (COTS) hardware and software for supporting the integration of new capabilities and interoperability between Apache helicopters and unmanned aerial vehicles (UAVs). The MCAP also served as a method for risk reduction for the Apache Block III program (Johnson 2006).

According to the Apache MCAP documentation, the team collected on the program, the development was scheduled to begin in April 2003 and was expected to be completed in two-and-a-half years at an estimated cost of $40 million. Its schedule was extended to June 2006 and its total cost grew to $50 million (Johnson 2006). The cost values for MCAPS discussed are notional and for simulation purposes only.

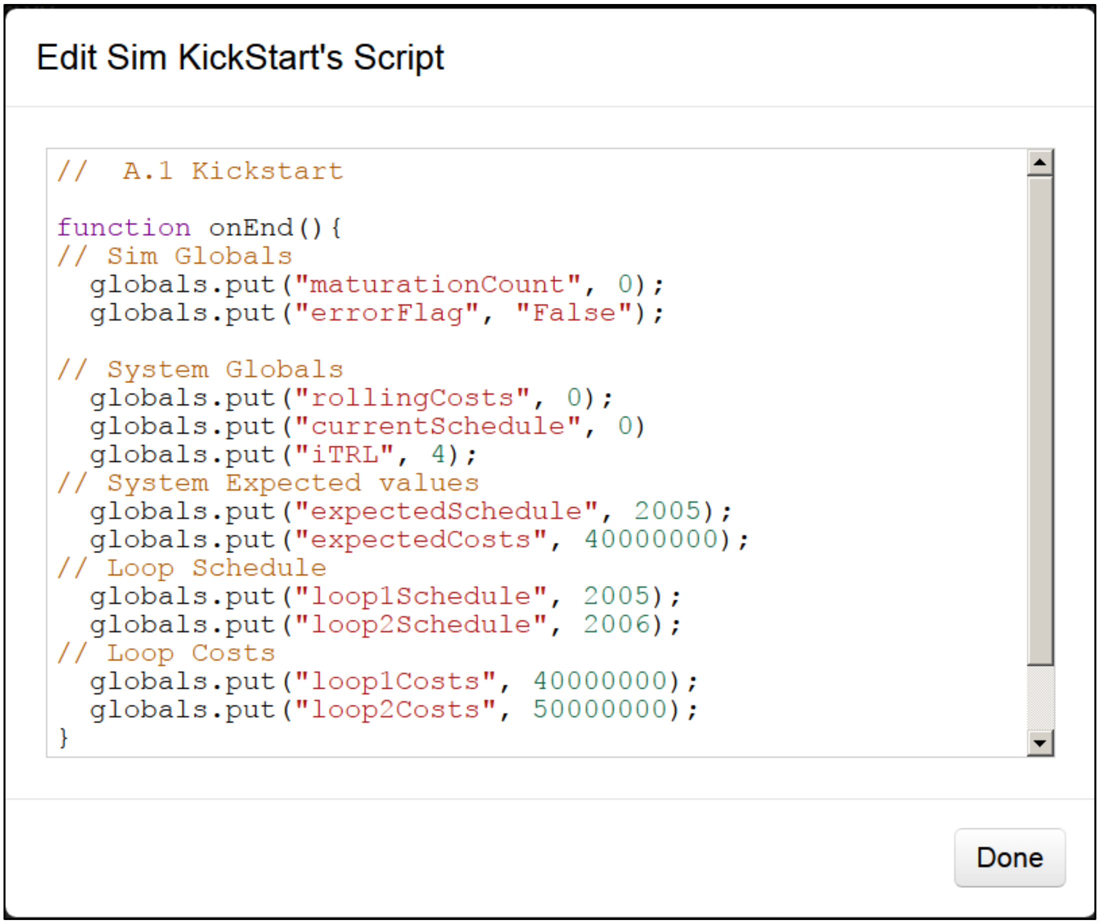
The input parameters for the Apache MCAP Use Case are shown in Table 20. The reader should note that the Apache MCAPS expected total cost, actual total cost and TRL program data were not accessible. The values utilized in this simulation for the MCAPS expected cost, actual cost and TRL are notional values and for simulation purposes only. The use of these notional values does not affect the ability to run the simulation. The notional values remain effective as inputs for evaluation of the function of the model.

Table 20. Apache MCAP Use Case Input Parameters

Parameter	Value
TRL	‘4’ *
Total Expected Schedule	‘2005’
Loop 1 Schedule	‘2005’
Loop 2 Schedule	‘2006’
Total Expected Costs	‘40000000’ ($40 million) *
Loop 1 Costs	‘40000000’ ($40 million) *
Loop 2 Costs	‘50000000’ ($50 million) *

* See note in final paragraph of this section for values explanation.

In order to input the Apache MCAP into the TDS simulation model, the team extracted the specific parameter values required by the model. The input parameters that were used for the MCAP Use Case are shown in Table 20. Figure 54 depicts the simulation global variables that are scripted as part of the Sim Kick-Start function. Upon completion of the simulation scripts, the team executed the model and the results of the simulation are shown in Figure 55.



Edit Sim KickStart's Script

```
//  A.1 Kickstart

function onEnd(){
// Sim Globals
  globals.put("maturationCount", 0);
  globals.put("errorFlag", "False");

// System Globals
  globals.put("rollingCosts", 0);
  globals.put("currentSchedule", 0)
  globals.put("iTRL", 4);
// System Expected values
  globals.put("expectedSchedule", 2005);
  globals.put("expectedCosts", 40000000);
// Loop Schedule
  globals.put("loop1Schedule", 2005);
  globals.put("loop2Schedule", 2006);
// Loop Costs
  globals.put("loop1Costs", 40000000);
  globals.put("loop2Costs", 50000000);
}
```

Done

Figure 54. Apache MCAP Use Case Model Scripts

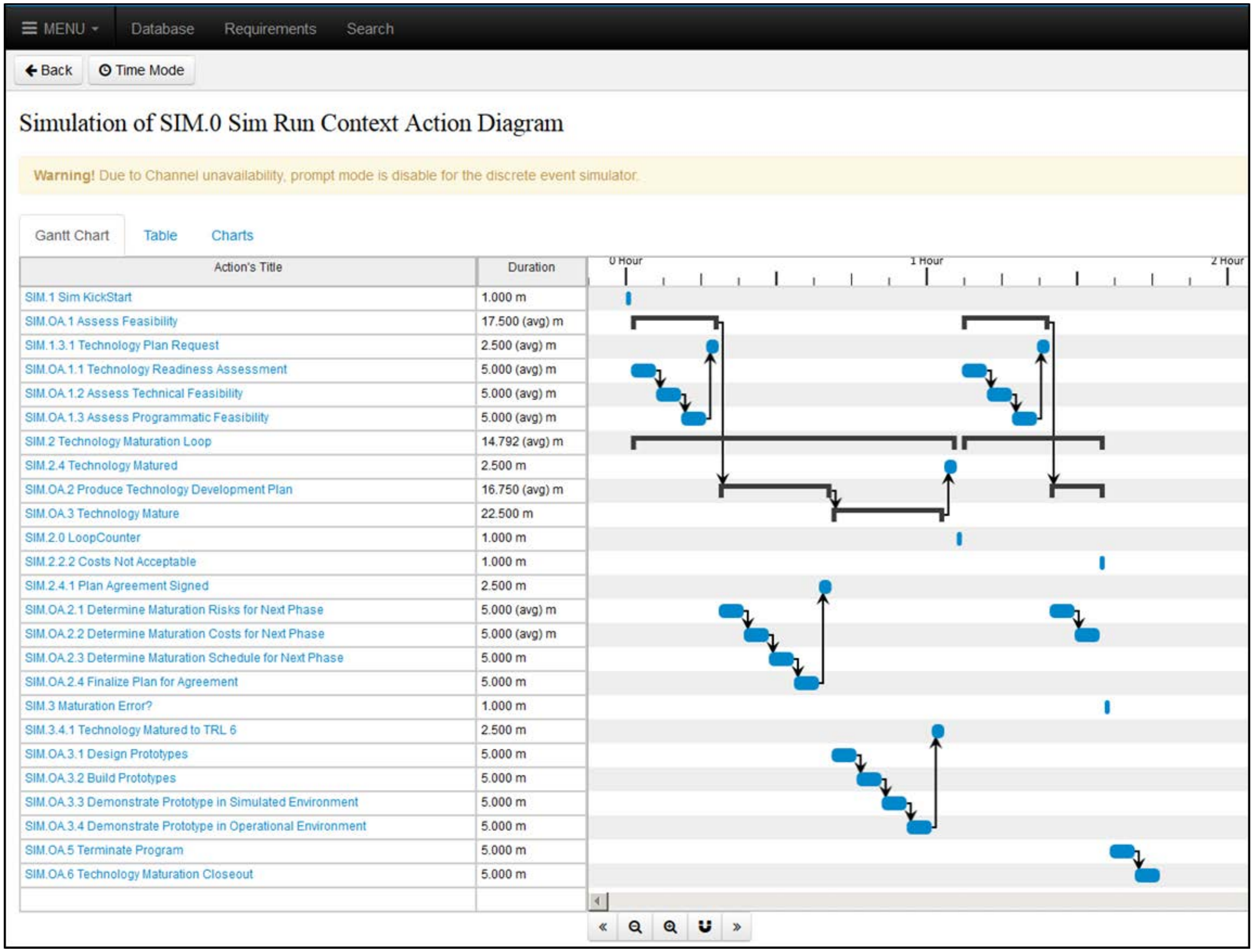


Figure 55. Apache MCAP Use Case Simulation Results

The simulation execution was expected to enter into the maturation loop and pass through the loop once. Upon entering the maturation loop the second time an error was expected to be triggered in function SIM.OA.2.2 Maturation Costs for Next Phase. The reason that this error was anticipated is because the total expected costs of $40 million was expended in the first loop so there would have been no more funding available to proceed through the second loop. The error message triggered the simulation to exit out of the loop and pass the message to the Terminate Program function. The Terminate Program function recognizing that the error flag had been set would send the simulation to the Terminate Program function and then finished up with the Maturation Closeout function.

Again, it is important here to highlight that one of the constraints of the simulation is that it can only provide a limited analysis based on the quality of the data that is provided into it. The reader should note that the Apache MCAPS expected total cost, actual total cost and TRL program data were not accessible. The values utilized in this simulation for the MCAPS expected cost, actual cost and TRL are notional values and for simulation purposes only. The use of these notional values does not affect the ability to run the simulation. The notional values remain effective as inputs for evaluation of the function of the model. Therefore, it is impossible to state unequivocally, that the TDS system approach would have resulted in a successful development of the Apache MCAP system. However, the resulting data shows that had the TDS system approach been used and resulted in similar results as the actual development, the TDS system would have been raising issues about the technology’s lack of mature ability much earlier in the process.

The original schedule showed that Apache MCAP was expected to have its technologies at a TRL of 6 by 2005 and within a budget of $40 million but the actual development that took place up to 2005 was only able to advance the technologies from a TRL of 4 to a TRL of 5 at a cost of $40 million. The TDS system has several locations throughout the process, that had it been utilized would have highlighted the issue much sooner than 2005. As an example during the planning phase of the TDS system, if the system had come back with a detailed cost and schedule that would have shown the development only reaching a TRL of 5 after having expended the entire budget and used the entire schedule, this would have raised a red flag. If the system had projected that TRL 5 would have been reached in 2004 yet the system was still going through maturation and had not matured to the level expected and as expected, this also would have raised a red flag.

Based on the data that was available and the current data checks that had been incorporated into the TDS simulation, the Apache MCAP use case simulation ran through the initial maturation loop and encountered an error during the execution of the loop 2 costs function because the combined total of the cost for loop 1 and loop 2 exceeded the total expected costs amount. This resulted in the simulation running the project termination function and finishing with the program closeout function. This behavior resulted as expected.

C. SUMMARY

In Chapter IV, the team developed the TDS executable model to simulate the operation of the TDS functions and to measure its performance in validating historical system development data. The model was validated and verified with multiple use case scenarios designed with expected outcomes and evaluating the models actual outcomes against the expected outcomes. The intent behind the model was not to show that the TDS approach was the only solution to system development but rather a proof of concept. The simulation demonstrated the TDS approach provides increased opportunities to track maturation progress and re-evaluate the state and progress of the program development. In Chapter V, the team provides a discussion of observations and conclusions with respect to the TDS concept and the original problem statement.

THIS PAGE INTENTIONALLY LEFT BLANK

V. SYSTEM ANALYSIS

According to Oliver, Kelliher and Keegan, system analysis is the study of the system of interest. The system of interest may be a product, a process, a business to be re-engineered, or a plan. System analysis is preceded by concept analysis that establishes the value of features of the system of interest to the organization, to its owners, and to users of the system. Analysis of the TDS concept was based on the use cases and simulations created to demonstrate the proof of concept in the previous chapter. The results of concept analysis are the initial information used for system analysis (Oliver, Kelliher and Keegan 1997).

A. SYSTEMS ANALYSIS OVERVIEW

Systems Analysis is a key step in any systems engineering process. The systems engineering process used for this project, as shown in Figure 1, contains a provision for reevaluating the system until the final recommendation is made. Every design concept required evaluation to ensure it satisfied the minimum requirements set forth at the beginning of the project and that the effective need has been met (Blanchard and Fabrycky 2011). Performing system design and analysis enabled the team to fully establish the context of the TDS and yield the details for decomposing the system into its constituent components. In order to progress to a suitable solution, the following key questions had to be addressed throughout the SE process:

- What decisions does the user need to make in order to accurately and effectively mature and transition technology?
- How should the TDS model be structured based on user needs and current DOD acquisition shortfalls?
- Is there a natural sequence that the model should follow to make maturing technology during this phase of acquisition more efficient, cost effective, and successful?
- How much detail is needed or can be provided within the constraints of a capstone project environment?

- What is the intended performance of the functions and how does this compare with the requirements of the user (Langford 2012)?

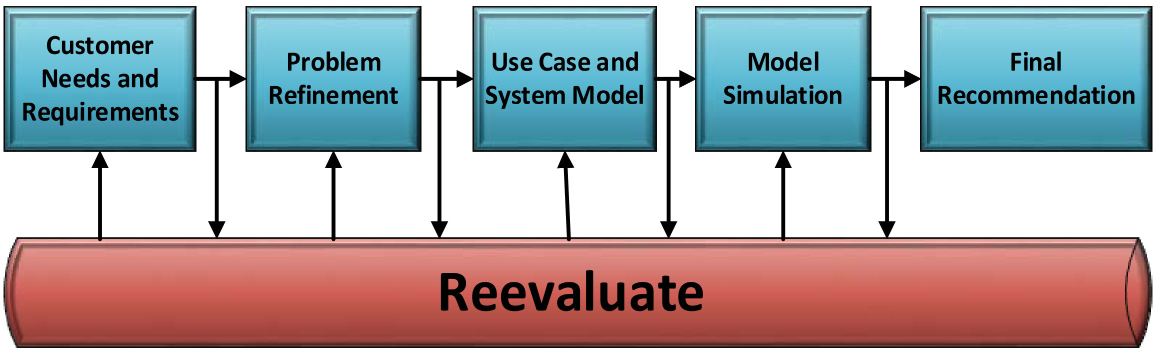


Figure 56. Tailored Systems Engineering Process (after Bahill and Gissing, 1998)

To trace the system design to the effective need of the user, an iterative process of analysis and evaluation was implemented. This approach was taken to define the problem in qualitative terms in sufficient detail to progress to the next step (Blanchard and Fabrycky 2011).

B. ANALYSIS OF THE TDS

To properly analyze and assess the TDS performance, its comparison with a known standard or benchmark was required. The effective need was selected to ensure the TDS met the intent of the project's fundamental approach of utilizing SE processes and techniques. Design choices were selected during the problem refinement and analysis stage that yielded the TDS processes and the order that these steps would be performed.

Verification of the TDS process required the development of functional models that were iterated, and continuously reviewed. Use cases were used to verify the core functions of the TDS and whether the processes being exercised would yield the expected results. Simulation results provided confidence that the functional model performed as expected. The team had the benefit of hindsight when reviewing the use cases. In most cases, the mistakes made during these acquisition developments were presented in the respective doctrinal source along with recommendations to overcome the acquisition shortfalls. The flaws that caused cost, schedule, or performance overruns along with the corrective actions provided were factored into the simulation in order to observe the effects realized when executed using the TDS model.

1. TDS Compared to the Effective Needs Statement

The revised and final effective needs statement for this capstone project is as follows:

> The DOD needs to change their current multiyear acquisition process in order to develop and provide weapon system capabilities to the Warfighter more quickly and support the Warfighter's need to adapt to rapid changes in threat, mission, and technological environments, within the constraints of controlling and/or reducing costs given fiscal instability, and providing solutions that are relevant and delivered in a timely manner (Erwin, 2013).

Table 21 was created to show how the TDS functions traced to the key elements of the effective need statement. Each column heading represents a critical part of the effective need statement. The Xs indicate a TDS function or sub-function that contributes toward meeting one of the critical aspects of the effective need.

Table 21. TDS Functional Traceability

	Stakeholder Needs			
TDS Functions	Develop and Provide Weapon Systems	Control and/or Reduce Costs	Provide Relevant Solutions	Timely
1.0 Assess Feasibility				
Technical Readiness Assessment	X	X	X	X
Assess Technical Feasibility		X	X	X
Assess Programmatic Feasibility		X		X
2.0 Produce Tech Development Plan				
Determine Maturation Risk for Next Phase		X		X
Determine Maturation Cost for Next Phase		X		
Determine Maturation Schedule for Next Phase				X
Finalize Plan for Agreement		X		X
3.0 Mature Technology				

TDS Functions	Stakeholder Needs			
	Develop and Provide Weapon Systems	Control and/or Reduce Costs	Provide Relevant Solutions	Timely
Design Prototypes	X			
Build Prototypes	X			
Demonstrate Prototypes in Simulated Environment		X	X	X
Demonstrate Prototypes in Operational Environment		X	X	X
4.0 Transition Technology				
Finalize Technology Transition	X			
Perform Technology Readiness Assessment		X	X	X
Transition Technology Artifacts	X			
5.0 Redefine/Terminate Program				
Program Determination	X		X	
Redefine Program Plan	X	X	X	X
Capture Issue Metrics		X	X	
6.0 Technology Maturation Closeout	X	X	X	X

The TDS will provide the capability to the DOD to develop and provide weapon systems to the soldiers while controlling and possibly reducing costs. Technologies and/or capabilities developed using the TDS process model will produce relevant weapon systems for the warfighter through detailed assessments, planning, and prototype demonstrations inherent to the model. Finally, the TDS will allow weapon systems to be developed in a timely manner.

Mature technologies are pivotal to developing relevant and timely weapon systems (GAO 2011). If the TDS is implemented as a complement to the DOD Acquisition construct, program offices will have an opportunity to successfully transition mature technologies and capabilities into formal system acquisition and development at MS B. It will also serve to identify knowledge gaps in technology assessments and alleviate the problems that arise when acquisition programs proceed with immature technologies.

2. TDS Compared to Current Acquisition Practices

To compare the TDS to the current and past prototyping practices, three acquisition programs were studied and their outcomes compared to the expected outcomes from the TDS. The acquisition programs reviewed by the team are described in detail in the follow on sections.

a. FCS Case Study

As previously discussed in Chapter IV, the TDS simulation of the FCS program would have recommended the program be cancelled or redefined due to cost overruns and disparate technology readiness levels of the systems being integrated. Additionally, FCS progressed beyond MS B in May of 2003, six years prior to being cancelled. At the time of MS B only 7 of 31 critical technologies were at a TRL of 6. The TDS system would not have allowed FCS to progress to MS B until these critical technologies were demonstrated at a TRL of 6.

b. VH-71 Presidential Helicopter Case Study

In 2009, the Navy's VH-71 Presidential Helicopter was cancelled after its budget increased from $6.5 billion to $13 billion and suffering a Nunn-McCurdy breach. In addition to the budgetary issues, there were also schedule and performance concerns (GAO 2011). At the point of cancellation, $3 billion had been spent and nine of the helicopters delivered to the Naval Air Systems Command. These nine helicopters were eventually sold to Canada for spare parts at a fraction of their cost (GAO 2011; Reed 2012). A GAO report on lessons learned from the VH-71 failure stated, "a primary reason for cost and schedule problems is too many technical unknowns and insufficient knowledge about performance and production risks." Two primary functions of the TDS are assessing the feasibility of a program and maturing the technology. The VH-71 program would have resulted in one of two likely outcomes if the TDS model were utilized. The first possible outcome is the VH-71 cost overruns and performance concerns would have been recognized more quickly and the program cancelled earlier, saving the DOD a portion of the $3 billion spent on the program. Second, the TDS would have allowed the technologies and capabilities needed for the VH-71 to mature to a TRL of 6.

In the latter case, the technical unknowns would have been eliminated and the performance and production risks reduced, providing the program a much greater chance of success.

c. Joint Tactical Radio System (JTRS) Ground Mobile Radio (GMR) Case Study

The DOD was striving for a joint tactical radio system that was software defined, reprogrammable, and could communicate with any other radio in the DOD. Despite the congressional attention provided to the program, JTRS was cancelled after years of technical failures and budget overruns (Gallagher 2012). JTRS suffered a Nunn-McCurdy breach in 2011 after research and development costs had grown by almost 70 percent in the period from 2002-2011 (Hoffman 2011). In a letter to the Senate Armed Services Committee announcing the termination of the JTRS GMR, Under Secretary of Defense for Acquisition, Technology and Logistics (USD [AT&L]) Frank Kendall stated, while explaining the reasons for terminating the program, that the technical challenges of the program were not well understood at the onset because of the immaturity of the technology (Kendall 2011).

One key purpose of the TDS is to mature technology prior to MS B. It seems likely that the JTRS GMR program would have never proceeded past the Assess Feasibility activity within the TDS model. This initial step in the TDS would have shown it to be very unlikely that the technology could have been matured to a TRL of 6 within the budgetary and schedule constraints. This would have forced DOD acquisition leaders to make decisions early in the development effort concerning their commitment to the JTRS concept given the cost and schedule required to develop the technology to a mature state for entrance into formal system acquisition.

C. TDS IMPLEMENTATION RISKS

The TDS prototyping process has been carefully researched and developed, however certain risks were identified and examined. Based upon the review of the TDS and supporting doctrinal sources, there are three main risks for implementing and operating the TDS prototyping process in the current DOD acquisition architecture.

First, users will proceed with the prototypes developed in the TDS process rather than using the prototypes to boost the development in the Engineering and Manufacturing Development phase of the life cycle. After a prototype has been demonstrated in an operational environment, users may have a tendency to view the prototype as the finished product (Weinberg 1991). This tendency can lead to good prototypes becoming fielded systems that offer very little value to the soldier. This risk must be mitigated by clearly managing expectations early in the TDS process. The technology development plan should include a section that clearly states that prototypes created for the purpose of maturing technology will not be ready for fielding and cannot be deployed (Plato 1995).

Second, there is a risk associated with the management of unrealistic expectations. After seeing a prototype demonstrated in an operational environment, there may be undue pressure to reduce cost and schedule estimates (Weinberg 1991; Plato 1995). The first few iterations of a prototype typically result in immediate high level results. A high level, crude prototype may demonstrate a concept, but it should not be misconstrued as a guarantee of project success in the TDS process or in the rest of the DOD acquisition development system (Plato 1995). Successfully maturing technology using the TDS process could be misleading. A change in the expected operational environment of a system can lead to the results of the TDS needing to be revalidated (Weinberg 1991). A potential mitigation tactic for this specific risk area is education of the users about the need to remain consistent in project focus and understand the limitations of the TDS process.

Finally, the possibility exists of resistance from the defense acquisition community due to increased cost and schedule requirements in the early phases of a capability development effort. This may occur in times of tight budgets when early prototyping, as described by the TDS, requires early investments in research and development creating difficult financial pressures (Borowski 2012). The costs related to implementing TDS, its risks, and justifying benefits are discussed in more detail in the following section.

D. TDS IMPLEMENTATION COSTS

Building prototypes early in the design process allows the cost and schedule risks to be fully understood by fostering a high level of technological maturity by the start of MS B. This has been documented as one of the indicators that can reduce the risk of system development in terms of cost, schedule, and performance (GAO 2013). While the TDS offers an effective way to discover and reduce risk, the process will raise costs. When the knowledge gained through prototyping is not available, technical risk is underestimated leading to increased project cost and schedule slips (Borowski 2012). The TDS will, initially, increase costs due to the mandatory prototyping of critical technologies prior to a MS B decision. In similar fashion, the more capabilities or technologies that are involved in the system, the more expense that can be expected (Borowski 2012). These increased costs for the acquisition phase between MS A and MS B, where the TDS process is implemented, will be offset and justified by the increased technical, cost, and schedule knowledge and reduced technical risk later in system development. Figure 57 demonstrates that as the probability of technical risk decreases, the cost estimate decreases.

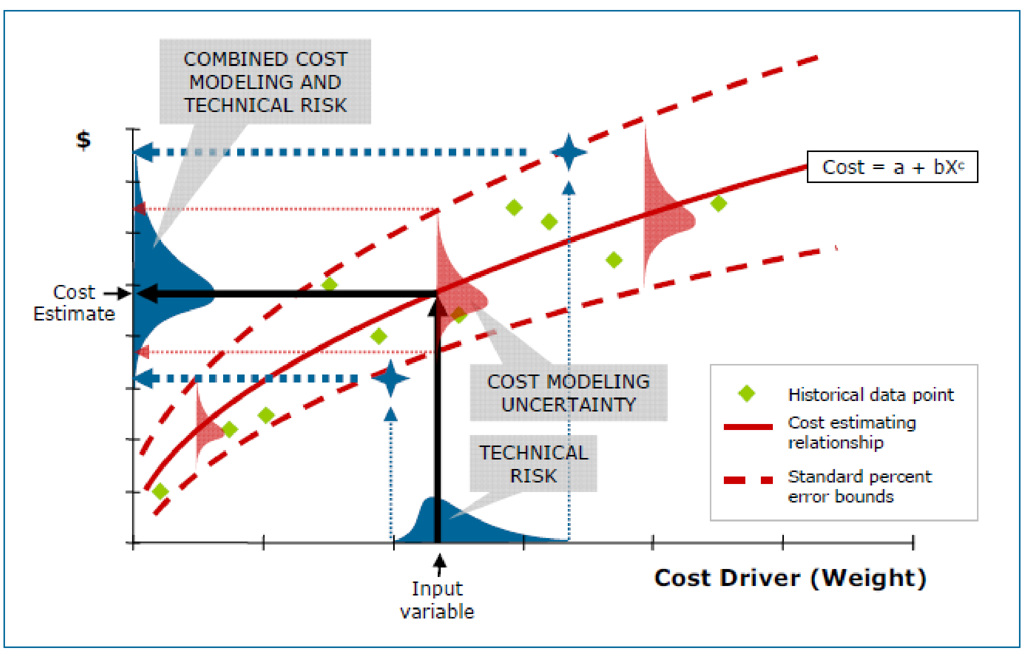


Figure 57. Cost vs. Technical Risk (from NASA 2008)

As the TDS matures technology and raises the TRLs, budget analysts can use the TRLs to estimate the technical baseline of each element in a system work break down structure and adjust the cost estimation accordingly. Although "actual" cost data for a system as it transitions through the TDS process does not yet exist, it is expected that technical risk will be reduced through the utilization of mature technologies. As a result, the overall cost of system development should be reduced.

In addition to cost savings due to more mature technologies early in system development, it is also anticipated that significant cost savings will be realized through the Assess Program Feasibility function of the TDS. The DOD has spent $50 billion over the past decade on programs that have ultimately failed (Erwin 2013). The Assess Program Feasibility function will ensure the technology attempting to enter the TDS model has achieved and demonstrated a clearly defined TRL and can be feasibly developed to a TRL 6, given the state of the technology and the programmatic, cost and schedule, constraints. If this function of the TDS had been in the DODI 5000.02 regulations over the past decade, it can be postulated that some portion of this $50 billion wasted on failed programs could have been avoided due to early recognition of the difficulty in technical maturation.

E. RESULTS

The TDS has a higher level of performance than current DOD processes in the critical areas of performance, cost, and risk. Table 22 uses an "X" to denote which system has the advantage in the specified metric. The justification for choosing one system over the other is presented in the following paragraphs.

Table 22. System Comparison

	Performance	Total Program Cost	Risk
DODI 5000.02			
TDS	X	X	X

In terms of performance, the TDS ranks far ahead of the current system. In the fall of 2013 the DOD released the Interim DOD Instruction (DODI) 5000.02. The Interim DODI 5000.02 updated the policies for the management of DOD acquisition programs. Like the TDS, the Interim DOD 5000.02 recommends prototyping to reduce risks and mature technology, however it tends to provide a structure that may not provide sufficient detail to improve success for every program. This loose construct has many useable features but lacks the direction that program managers and acquisition professionals need to adequately uncover gaps in technology early enough to allow detailed technology development planning. Likewise, the Interim DOD 5000.02 differs from TDS in that it provides exceptions that allow the requirement for prototyping to be waived during the TMRR phase between MS A and MS B (USD [AT&L] 2013). The TDS requires prototyping to mature technology to at least a TRL of 6 prior to MS B. The TDS provides a detailed functional architecture for developing the technology through prototyping, testing, and assessments. The Interim DODI 5000.02 does not provide any instruction on how prototyping is to be performed, nor is a required level of technology maturation specified. The clear requirement for prototyping to be performed and a specific TRL to be obtained clearly raises the performance of the TDS over the Interim DODI 5000.02 process.

As discussed previously in this chapter, the TDS would provide long term cost savings through the utilization of mature capabilities and technologies for DOD system development. The Interim DODI 5000.02 instructions allow for waivers for cost savings, but these cost savings can be lost later in the program if the result is an immature technology. According to extensive research performed throughout the execution of the capstone project, the DOD routinely accepts high levels of technology risk at the start of major acquisition programs (GAO 2006). A defined phase for technology development and transition has been identified but the construct has not been adequately defined to realize the benefits. These shortcomings have contributed greatly to the DOD’s poor cost and schedule outcomes (GAO 2006). For total program cost, the TDS is preferable due to the upfront cost providing the opportunity for the use of mature technologies later in the program. These considerations provide the TDS with a cost advantage over the Interim DODI 5000.02 process.

Both the Interim DODI 5000.02 and the TDS offer methods for reducing risk. In addition, they also suggest the use of TRLs, though the Interim DODI 5000.02 does not require a specific TRL be obtained prior to MS B. Since the TDS is intended to be used in concert with current DOD acquisition policies, the requirements of MS B will apply to both systems equally. MS B requires that risks show adequate mitigation has taken place. While the Interim DOD 5000.02 provides a statement that says risks must be addressed, there are no regulatory statutes that provide a benchmark, such as a specific TRL. The TDS has specific TRLs to ensure the critical technology is matured and provides a specific process for maturing the technology. The specific processes and mature technologies provided in TDS have the potential to offer more risk reduction than the Interim DODI 5000.02 process.

F. SUMMARY

Systems analysis, from a systems engineer's point of view, can be stated in the most simple terms as (1) describing the problem in sufficient detail to effectively support the development effort; (2) designing an alternative, or set of alternatives, that reflect the functional architecture and is responsive to the user need; (3) verifying that what was developed and deemed the system solution matches the requirements of the stakeholders; and (4) validating that what was developed can be traced back to the problem and needs identification (Langford, 2012). The capstone team sought to follow these general guidelines to move the project from problem to solution.

The team entered into the conceptualization stage focused on two primary activities. The first activity centered on defining the problem faced by the stakeholders, and thus, translating this problem definition into an effective need. The second focused on developing a concept of operations (CONOPS) in order to set the preliminary course for exploring the problem, need, and solution space (Langford, 2012).

Entrance into the Needs and Requirements and Problem Definition stages of the SE process included many activities that were done recursively through analysis, verification, and evaluation. In this stage of the process, the team sought to transform the CONOPS into requirements at appropriate levels of detail to produce a design that could be evaluated for risk, applicability, cost impact, and overall performance. Detailed design commenced with functional architecting, modeling, and simulation. As discussed in Section V.A.2, the TDS was modeled extensively and verified using several case studies for relevant applicability. The goal was to find current prototyping activities and compare their actual performance and outcomes with the expected outcome should those same prototyping activities have been completed using the TDS model. The results of the simulation exercises have shown very promising trends of maturing technologies as recommended by GAO and other independent researchers.

Due to the broad applicability of the TDS, the risk identification associated with the implementation of TDS required careful research and analysis. The TDS was envisioned and developed to be applicable across the DOD to any program office from the major armed services. There were three TDS implementation risks discussed in this section: early prototype acceptance, unrealistic expectations, and DOD resistance. The cost of implementing the TDS model into the DOD acquisition construct was given special attention and broken out separately from the previously mentioned implementation risks. It has been suggested, based on careful research, in GAO reports from 2006, 2010, and 2012, that increased cost in early system development can provide mature technologies that, in turn, reduce the overall cost of system development. With budget concerns rising, however, this upfront cost increase poses a risk to programs in the early phases of development. Even though the TDS model will increase costs between MS A and MS B, it will also significantly reduce the technical risk of proceeding into formal system development with immature technologies. As shown in Figure 57, as the probability of technical risk decreases, the cost estimate for system development decreases.

This section on systems analysis provided an overview of the activities performed and how the team applied it to this capstone project. There is no single method for all problems, therefore, the team chose an SE method and tailored that method to meet specific needs. The general principles of SE and trusted techniques for problem solving using systems analysis have guided the TDS development effort. These systems analysis techniques provided the benefit of greater insight into the problem being researched, useful decision making guidelines, and a structured application for progressing from problem identification to solution implementation (Beimborn 2003).

THIS PAGE INTENTIONALLY LEFT BLANK

VI. RESULTS AND CONCLUSION

In this capstone report, the current state of early system prototyping within DOD acquisition was detailed including a brief history of weapon system programs frequently entering into system development with immature technologies. Extensive research revealed the breadth of the problems facing the DOD acquisition community and were detailed in numerous reports from multiple sources, including the GAO, RAND Corporation, academia, and the military services.

The project addressed the need for the DOD to change its current acquisition process as it relates to early system prototyping and technology development. From the research, it was determined that the DOD needs a prototyping process that is based on organized principles that are standardized and repeatable. This process also needs to include synergistic programmatic and technical methodologies as well as, success metrics, to support effective early system prototyping and technological development of DOD weapon systems (Erwin 2013; Lane et al. 2010).

A. RESULTS AND RECOMMENDATIONS

The long term objective of this capstone effort was to produce an extensible technology assessment and development method that can be applied to technologies that have previously achieved a TRL 4 prior to entering the TDS process. The research and development associated with the project sought to overcome the DOD's challenges by:

1) Effectively defining a standard technology assessment method in order to accurately and objectively determine the strengths and weaknesses of an incoming technology.

2) Identifying the appropriate planning structure to develop an accurate and feasible programmatic and technical plan for maturing and transitioning the technology.

3) Introducing the opportunity within the structure of the model to either redefine the development plan or terminate the development effort should it become necessary.

Due to academic time constraints, the capstone team was directed, consistent with the priorities of the NPS, toward a subset of the DOD Acquisition life cycle. This pilot process model only focuses the methods necessary to assess a technology entering MS A, develop the technology, and transition the successfully demonstrated technology into formal system acquisition at MS B. The challenges were documenting the overarching definition of prototype and prototyping as it relates to the DOD as a whole, identifying the current prototyping environment and how it is used to reduce technical risk, identifying the appropriate set of activities needed to successfully develop and transition technology, and compiling the root causes of weapon system development failures and the creating a process model that overcomes these gaps in technology development.

The team determined that the DOD needed a standardized and tailorable prototyping process that provided organized principles, synergistic programmatic and technical methodologies, and success metrics in order to support effective early acquisition prototyping and technological development. In response to this need, a solution-neutral, process model was developed to assess the technical and programmatic feasibility of developing the technology, along with the planning, iterative technical reviews, and transition strategies to ensure successful future development once the system entered formal acquisition at MS B. The model was developed with a multidisciplinary team using SE processes learned throughout the course of study.

During the capstone process, the following key contributions were provided as deliverables to NPS and RDECOM:

- Repeatable and extensible process model for addressing the DOD early system prototyping challenges
- Innoslate Model executable reference architecture with bi-directional links between system entities and attributes
- Innoslate executable simulation model

- Draft Model Taxonomy for the Technology Development System
- Resource Data

The TDS represents a means to provide new and/or improved acquisition processes, specifically within the TMRR Phase, to facilitate delivery of a flexible solution to the user that meets mission needs. The set of solutions that comprise the TDS will satisfy the DOD’s need to comprehensively and objectively assess program feasibility, plan for technological development, mature the technology, transition the technology into formal system development, while also providing the necessary iterative loop that allows a redefinition or termination of the effort should it be deemed appropriate. Since the TDS is model based and solution-neutral, it can be extended to any similar development effort across the DOD.

B. RESEARCH QUESTION RESULTS

According to a 2013 GAO report titled, “Defense Acquisitions: Assessments of Selected Weapon Programs,” there is a positive trend over the last four years of DOD program offices demonstrating higher levels of technical knowledge at key decision points. Many programs, however, are still not fully realizing success in terms of cost and schedule versus performance. Of the thirty-two programs that provided GAO researchers with technology maturity data, five had been deemed fully mature when they began development (GAO 2013). For those five programs with self-ascribed fully mature technologies, less than one third had stable designs at critical design review (CDR) (GAO 2013).

Numerous GAO reports have concluded that most DOD programs proceed with a low level of technological knowledge resulting in cost and schedule increases. Only 16% of programs achieved mature technology at MS B. Programs that did not have mature technologies upon entry into MS B averaged 32% cost growth and a twenty month schedule delay (Gordon 2008).

Throughout the capstone project process, research into successful and unsuccessful DOD weapon system programs revealed a common theme: knowledge supersedes risk over time. Positive acquisition outcomes require knowledge-based approaches before significant development commitments can be made. Successful programs have anchored their approach in attaining and demonstrating technical knowledge at critical decision points in the process (GAO 2013).

The GAO has identified three key knowledge points that are essential during the acquisition cycle (GAO 2006). Knowledge points two and three do not apply to the acquisition phase that is the subject of this capstone report and outside of the scope of this effort. Knowledge point one aligns with the start of MS B, referred to as the Engineering and Manufacturing Development Phase. Achieving a high level of technological maturity by the start of MS B is one of several indicators that reduce the risk of system development in terms of cost, schedule, and performance (GAO 2013). The technologies that are being transitioned into this phase need to have successfully demonstrated operation in a relevant environment.

To guide the project, research questions were defined and answered by the team that supported the final conclusions and recommendations. The research questions are listed:

- How is prototyping defined with respect to DOD acquisition?
- How does the DOD currently use the prototyping process to reduce technical risk?
- Why is early acquisition prototyping not currently realizing success in the DOD acquisition process?
- What activities are performed in early acquisition prototyping?
- What metrics can measure prototyping success?

1. How is prototyping defined with respect to DOD acquisition?

One tool that is prevalent within the DOD acquisition framework is prototyping. Using early prototyping during development can reduce technical risk, refine requirements, and validate design and cost estimates (GAO 2013). The blanket statement, “we can use prototyping,” however, is not as simple as it sounds. There are many layers, both technical and programmatic, that form the context for prototyping a particular technology or capability. One key aspect of realizing successful demonstration through prototyping is to have a common understanding among all vested program stakeholders as to what is meant by the terms “prototype” and “prototyping.” The DOD has been performing some level of prototyping for fifty years or more, however, the definitions for these terms remain varied and undefined (Borowski 2012). After researching and analyzing many different definitions for these terms, as discussed in Appendix B, the definitions provided by Samuel Borowski during a 2012 Defense Acquisition University (DAU) Symposium captures the breadth and depth of the two terms most completely:

- A “prototype” is a test article designed to demonstrate areas of high technical risk that are essential to system success. A prototype need not be a full system, but, in scope and scale, it is tailored to accommodate a series of decisions, and as such, can represent a concept, subsystem, or end item according to the decisions to be made. Rather than reflect the final design, prototypes are built with the expectation that, as decisions are made, change will follow (Borowski 2012).
- “Prototyping” is the practice of testing prototypes, of appropriate scope and scale, for the purpose of obtaining knowledge about some requirement, capability, or design approach. The knowledge obtained informs decision-making, the output of which results in some degree of change. The degree of allowable change is bounded, in inverse proportion, by the scope and scale of the prototype (Borowski 2012).

2. How does the DOD currently use the prototyping process to reduce technical risk?

It is accepted that prototyping is performed to reduce the technical risk associated with a development effort. In order to form the basis for developing a system model to improve prototyping in a DOD acquisition construct, the capstone research team needed to identify the current DOD prototyping environment along with how the prototyping process is used to reduce technical risk. The DOD has certainly recognized the need for more structure in the early phases of acquisition, as well as, identifying the positive results due to prototyping (Dahmann and Bhatti 2008). The WSARA, enacted in 2009, requires the DOD to produce prototypes before a design is selected for further development prior to MS B, unless a waiver is granted by the MDA (Sullivan 2013). Even though early system prototyping has been recognized, required, and considered a best practice in a multitude of sponsored and independent reports, the DOD still delegates responsibility over the prototyping process, as well as the decision as to whether prototyping is needed, down to the Program Manager level. This has resulted in ad hoc prototyping occurrences and disparate methodologies among the military branch acquisition constructs. There is no formal prototyping process model that has been accepted by the Program Offices within the DOD. Therefore, there is no clearly defined prototyping method used by the DOD to reduce the technical risk of acquisition programs.

3. Why is early acquisition prototyping not currently realizing success in the DOD acquisition process?

In a study titled, "Stronger Practices Needed to Improve DOD Technology Transition Processes," the GAO identified three techniques used by industry for developing and transitioning technology: Strategic Planning preceding technology development, Gated Management Reviews, and Corroborating Tools such as transition agreements. The results of this study indicated that the DOD lacked the breadth and depth of any of these techniques and, therefore, concluded that many of the cost and schedule overruns on major weapons acquisition programs could have been prevented. The DOD routinely accepts high levels of technology risk at the start of major weapon acquisition programs (GAO 2006). It would stand to reason that if prototyping is good enough to support a production decision, why not use it earlier in the acquisition process to justify a formal program start at MS B? Critics would argue that, while prototyping may provide value, there is too much change early in the life cycle to make prototyping worthwhile (Borowski 2012). Many of the issues that have plagued early acquisition prototyping are the result of a lack of understanding between the technology and programmatic communities in reference to technology and performance objectives and operational concepts prior to MS B (Sullivan 2013). Practitioners within the acquisition community have not understood how early system prototyping should be approached. Adding to the difficulty, policy makers have struggled to organize principles and methodology around prototyping from which to elicit better outcomes (Borowski 2012). This project was initiated to help address this void by identifying, documenting the best practices from different technology and prototyping models, assembling a tailorable methodology for early system prototyping, and maturing technologies prior to formal system development at MS B.

4. What activities are performed in early acquisition prototyping?

Prototyping is a useful tool for any system that requires the demonstration of a new technology as part of its development. Prototypes built prior to the MS B phase of procurement are generally intended to prove a new technology or set of technologies and demonstrate a basic approach for their implementation into a developmental system (Cate 1997). These prototype artifacts allow acquisition decision makers to determine if an approach to a system can be further developed in the acquisition phases following MS B. Over the course of researching prototyping methods within the DOD, as well as industry, the differences in systems that utilize prototyping, management styles, or process structures do not seem to be major drivers in prototyping strategies (Drezner 1993).

The prototyping activities that are performed prior to MS B are widely varied and dependent on the programmatic goal that served as the impetus to prototype in the first place. Prototyping strategies are largely based on the different uses that prototypes serve. A technology demonstration prototype can be used to "verify and reduce the technology risk," evaluate operational concepts, or provide alternative choices (Drezner 1993). Our goal was to create a framework that includes most of the critical characteristics that define prototyping while keeping it simple enough to be useful in decision making. The system model that was created is based, in large part, on timing, degree of risk, and program goals. Timing simply refers to the phase where prototyping occurs; in our case, prior to MS B. Degree of risk directly relates to the level of technological maturity on which to base the prototyping plan. The programmatic goals are more difficult to represent because they can be varied among organizations or programs. There are multiple categories that can represent program goals: technology viability, technology demonstration, system design/performance, or operational system upgrade among others.

Developing generic strategies for prototyping is a highly subjective undertaking. Each program is unique with its own set of circumstances for developing prototypes. Instead of providing a "canned" set of activities for performing prototyping, the capstone team developed a general framework for ensuring success that includes assessing the current state of technological maturity, planning and documenting the prototyping effort, and generating the required documentation and deliverables required to satisfy DOD 5000 guidelines. Prototyping strategies within this framework will be focused on key risks and uncertainties within the particular program. An effective prototyping strategy should be tailored by the model provided in this report as well as program specific constraints and goals.

5. What metrics can measure prototyping success?

Before we can fully understand what metrics support prototyping success, we must first identify the root causes of risk and failure. The following, compiled from extensive research, have been identified as the primary drivers for cost/schedule growth for DOD Programs (Azizian et al. 2011):

- Unrealistic performance expectations
- Immature technologies
- Excessive integration risk
- Unanticipated design changes
- Poor program management
- Evolving requirements
- Rapid technology obsolescence

As stated earlier, the GAO, in a series of reports from 2000 through 2013, have highlighted several focus areas in the Technology Development phase to support more successful system developments. The GAO recommends an enhanced emphasis on technology maturity, systems engineering, and system/subsystem prototyping (Azizian et al. 2011). The lower the level of technology readiness, the more ground must be covered to bring the technology to the point that it can meet the intended cost, schedule, and performance requirements with little risk to the program (GAO 1999).

The development and execution of any prototyping process cannot be conceived or completed in a vacuum. There are many factors that contribute to successful prototyping. Consideration must be given to the acquisition phase entry point, the initial maturity of the technology to be demonstrated, and the intended outcome of the prototyping activities. Key engineering activities that surround prototyping in the context of the particular acquisition phase must be identified, agreed upon, and planned. Some of the specific activities that are key to reaching a successful MS B are laboratory evaluation of components, relevant environment evaluation of components, system/subsystem prototyping, and a TRA. All of the aforementioned activities are ineffective if other recommended systems engineering activities are not implemented in parallel, for instance documentation and planning (Azizian et al. 2011). It is important to develop a plan and follow the plan. It has been mentioned in numerous GAO reports that many acquisition programs do not implement TRA enabling activities, therefore, they may be advancing through the stages of acquisition with crippling technology knowledge gaps (GAO 2013; GAO 2006).

Measuring the success of a prototyping effort is not an easy undertaking. Scoring the success depends on many things, but most of all, it depends on what the technology and programmatic communities choose as the criteria for success. Success depends on one's perspective – the engineer may define success as limiting the amount of prototype revisions between the initial prototype and the prototype transitioned into MS B, the project lead may measure success in that the effort does not over-run cost and schedule. Success is a very subjective metric that must be documented among the stakeholders of the effort. This is why there must be cooperation and planning among the stakeholders. A team working toward disparate measures of success will be impeded and the project is at risk of not advancing (Kowal-Jurgens 2011). Organizations must partner with their technology counterparts to determine what and how to measure.

While metrics for success in a prototyping process are vague, the team did develop metrics that can be used to measure the effectiveness of an organization using the TDS. These metrics offer a way for an organization track for every function in the TDS. If these metrics, or evaluation measures, are tracked and the objective levels met, the organization will be successful in implementing the TDS and maturing technology. These metrics are discussed in detail in Chapter III and are listed in Table 23.

Table 23. TDS Metrics

Functional Trace	Evaluation Measure	Measure	Units	Classification	Threshold	Objective
1.1	Accuracy of TRL	MIB	%	Proxy/Constructed	95%	100%
1.2	Accuracy of Technical Feasibility	MIB	%	Proxy/Constructed	95%	100%
1.3	Accuracy of Programmatic Feasibility	MIB	%	Proxy/Constructed	95%	100%
2.1	% of Risks Identified	MIB	%	Direct/Constructed	95%	100%
2.2	Accuracy of Cost Prediction	MIB	%	Proxy/Constructed	80%	90%
2.3	% of Projects that Meet Schedule	MIB	%	Proxy/Constructed	95%	100%
2.4	% of First Time Signatures of Final Version	MIB	%	Direct/Natural	97%	100%
3.1	% of Design Requirements Met	MIB	%	Direct/Natural	80%	100%
3.2	% Design Specifications Met	MIB	%	Direct/Natural	80%	100%

Functional Trace	Evaluation Measure	Measure	Units	Classification	Threshold	Objective
3.3	% of Performance Requirements Demonstrated	MIB	%	Direct/Natural	90%	100%
3.4	% of Operational Performance Criteria Demonstrated	MIB	%	Direct/Natural	90%	100%
4.1	Time to Complete Customer Documentation	LIB	Time	Direct/Natural	6 Weeks	3 Weeks
4.2	Accuracy of TRL	MIB	TRL	Direct/Natural	95%	100%
4.3	% of Project Artifacts Without Rework	MIB	%	Direct/Natural	95%	100%
5.1	Accuracy of Determination	MIB	%	Direct/Natural	95%	100%
5.2	% of First Time Signatures of Final Version	MIB	%	Direct/Natural	97%	100%
5.3	% of Termination Issues Identified	MIB	%	Proxy/Constructed	95%	100%
7.1	% of Successful System Executions	MIB	%	Direct/Natural	95%	100%
7.2	% of Satisfied Customers	MIB	%	Proxy/Natural	95%	100%

C. CONCLUSION

The DOD should mandate all contractors and defense agencies performing prototyping between MS A and MS B to implement the TDS. The TDS will ensure that programs that reach MS B have technologies that have been matured to a TRL of 6. This will bring the DOD in line with the GAO’s recommendation that acquisition programs use mature technologies that are available for immediate use (GAO 2006). A failure to implement the TDS will allow the DOD to continue down the path that has been littered with billions of wasted taxpayer dollars and failed programs that provide no benefit to the warfighter (Erwin 2013).

D. FUTURE WORK

The capstone team found that the identified proof of concept model, provided as a culmination of the group's effort through three academic quarters, will be of great usefulness and reasonable fidelity given the resource and time constraints. It should be pointed out, however, that this is a notional model that has not been validated in an operational setting. These constraints limited the possibility to develop, simulate, and analyze all desired aspects of the Technology Development System model. There are several areas suggested by the team as focus areas for future capstone projects.

- Function 1.1, Technology Readiness Assessment, and Function 4.2, Perform Technology Readiness Assessment, should be further decomposed to determine the best method for assessing technology. Using TRAs to assess and validate TRLs is clearly defined and remains a well-accepted method within the DOD (ASD [R&E] 2011). TRLs, however, when used as a unit of measure to validate technological maturity or viability, are not without issue. TRLs are the standard for determining whether a technology is sufficiently mature to be incorporated into a system. This concept is useful; however, it does not address risk and obsolescence factors associated with maturing a technology to the desired end state needed for transition into formal system development (Valerdi and Kohl 2004). Further work should be conducted to either identify or develop a well-defined, holistic process for evaluating a technology that includes an assessment of its current technical readiness level (TRL), as well as, its present and future risk and obsolescence issues.
- Increase the current simulation model fidelity to improve relationships where necessary and replace notional parameters with actual program statistics for model validation.

- Increase the fidelity of the current model, with specific documentation requirements and activities, to align the lower level functions within the system to the DODI 5000.02 acquisition life-cycle model.
- Perform comparison simulations for the TDS and DODI 5000.02 acquisition models to enable analysis and determination of the best model for technology development during the TMRR phase of acquisition.
- Train systems engineers and acquisition professionals to use CORE, Innoslate, ExtendSim, and other modeling and simulation tools in order to achieve better system functionality by building a complete and consistent set of measurable requirements.
- Finally, work should be continued to improve the DOD prototyping process between MS A and MS B. This report has presented a possible framework for improving early prototyping in DOD prototyping and the team encourages all stakeholders to begin taking steps to implement this recommend framework.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A. RISK MANAGEMENT PLAN

A. INTRODUCTION

This RMP was developed using DOD's (2006) *Risk Management Guide for DOD Acquisition* and student team work from the NPS System Engineering Program Management course assignment work material. Each team member took the initiative to identify, analyze, track, and mitigate risks to ensure that the project would remain on track to meet the specified goals (DOD 2006). Risk management efforts began early in the SE process and phases of the project through the documentation of technical and non-technical risks for each development phase. These risks were modified, supplemented, and closed as the project progressed through each phase of the systems engineering process.

B. PURPOSE

The RMP describes how the project team integrated a comprehensive and proactive risk management process into the overall project management process. It details the risk management policy, definitions of key concepts and terminology, tiers of risk management, levels of risk, and risk management activities. It defined the process used to identify, analyze, track and mitigate and/or eliminate events and conditions that may have adversely impacted the project later in the life cycle.

C. SCOPE

The RMP describes the responsibilities and the processes utilized by the team to manage project risks. The status of project risks and their associated mitigation and contingency plans are documented in the risk management database, briefed to the team and NPS advisors every two weeks.

D. RISK

According to the DOD, a risk is a potential problem or event that has a negative impact on the project, that has not yet occurred. By this definition, if the event or problem is occurring or has already occurred, it is no longer a risk, but rather it is an issue (DOD 2006).

E. ISSUE

Once a risk event has occurred, it becomes an issue and the contingency plans are executed (DOD 2006).

F. ROLES AND RESPONSIBILITIES

Program Manager (PM)

- Assigns the Risk Manager (RM)
- Coordinates with the RM and risk owners to ensure that the risks are prioritized
- Receive appropriate level of management attention
- Are properly resourced with personnel and mitigation plans are executed
- Reports on risks to the NPS chain of command as required
- Verifies that risk management is integrated into all project activities
- Coordinates with the RM to consolidate the individual risks to determine the projects overall schedule, and performance risks

Team Member(s)

- Identify new program risks and their consequences
- Assess the severity of the consequences
- Determine means to mitigate risks
- Develop contingency plans
- Implement mitigation and contingency plans
- Update information on the mitigation status and the impacts of risk to reflect changes as they occur

Risk Manager (RM)

- Leads the management process

- Coordinates with the risk owners and the PM to ensure that the risks are prioritized
- Receive appropriate level of management attention
- Are properly resourced with personnel, and that mitigation plans executed
- Ensures a disciplined, repeatable risk management process is executed
- Ensures risk reports are available for each meeting or event where risks will be discussed
- Monitors the planning activities to ensure that they are consistent with this RMP
- Revises this plan as required to reflect any substantial changes in the risk management processes
- Authorized to add new risks and to close LOW risks. Obtains Team approval to close MEDIUM and HIGH risks
- Maintains the Risk database, including updating, compiling, analyzing and organizing risk data

G. RISK MANAGEMENT STRATEGY

The risk management strategy provided the project with a consistent plan to mitigate the probabilities and consequences of serious issues when possible; established pre-planned contingency plans to address risks when prevention was not feasible by providing the team with complete and current information to make informed decisions. In addition, the strategy established risk management into the daily activities and periodic risk management reviews. The strategy began with a baseline risk assessment that was conducted early in the project to define risk throughout project development (DOD 2003).

H. RISK MANAGEMENT PROCESS

The team utilized the organized methodology identified in the DOD (2006) Risk Management Guide for Department of Defense (DOD) Acquisition, as depicted in Figure 58, to continuously identify, analyze, mitigate, and track unknowns that could have adversely impacted the project. This process was iterative requiring project personnel to reexamine existing risks, make necessary modifications to existing risk mitigation and contingency plans, and track the status of mitigation efforts and related project events. The identification of new risks were addressed during team meetings as the project progressed. The risk management process continued throughout the life cycle of the project (DOD 2006).

Implementation of this process was intended to achieve:

- Effective communication and coordination
- Complete and current documentation
- Early identification and analyses of risks
- Early implementation of mitigation efforts
- Continuous monitoring of project, risk status and reassessing risks
- Continuous improvement

Figure 58. Risk Management Process from (from DOD Risk Management 2014)

I. RISK IDENTIFICATION

Risk identification was the first and most critical step in the risk management process. It required capturing a statement of what could have gone wrong in the future (the risk or risk event) and the circumstances in which it could occur (context of the risk). Once a risk occurred, it became an issue that the team had to address and mitigate.

J. RISK ANALYSIS

The risk analysis was conducted as a team effort to identify the risks associated with the project and future implementation plans. A root cause is the basic underlying identified defect causing the possible risk event or issue. If multiple-issue underlying identified defects exists then the root cause encompasses all of the defects in the total solution set. Methods, such as brainstorming, the Ishikawa (or fish) diagram and the 5 Whys, are some of the methods used to identify root causes. The team selected brainstorming as a method for determining the impacts of risk events (DOD 2006). The team collected sufficient information about each risk event and its context to:

- Determine root causes
- Define/refine cost, schedule, or performance impacts
- Determine the timeframe that the risk event can occur within
- Categorize the risk as a schedule or performance risk
- Determine the consequences should the risk become an issue
- Determine the probability that the risk event will occur (become an issue)
- Assign the risk rating (high, moderate, or low) based on the probability and consequence associated with the risk
- Update the risk description to ensure it clearly identifies the risk and the context of the risk

K. IMPACTS

The impacts for the project and system of interest risks were subdivided into performance, schedule and cost categories. Risks associated with future implementation of the system of interest will be identified and the impacts described in the report using research and the method identified in this plan. The performance impacts affected operational, technical, production, supportability and management requirements such as:

- Technical Performance Measures
- Reliability, Availability and Maintainability
- Interface Compatibility
- User Acceptability
- Producibility and Quality Control
- Configuration Management
- Testability
- Staffing Levels
- Personnel Qualifications/Experience
- Management Processes, Planning, and Documentation
- Safety

Schedule impacts may affect project milestones, including significant accomplishments and/or delay in deliverables. Because this project was an academic exercise, there were no cost impacts associated with its development; however, the cost to the DOD was considered when the project solutions were analyzed.

The risk description was refined to clearly capture the risk in terms of:

- Risk Event A occurrence that negatively affects the project or program.
- Risk Context (conditions – what, why, where when, how – that must exist for the risk to occur, including the root cause)

- Risk Timeframe - based on the event and impacts the identification of the latest date that the risk could have occurred
- Impacts on the project in terms of schedule and/or performance if the risk event was realized
- Determine Consequence or Impact Level

The impact level was based on the severity of consequences if the risk event had occurred. The Project utilized the five levels of impact as described in Table 24.

Table 24. DOD Levels and Types of Consequence Criteria (from Hoeferkamp and Zsak 2007, 26)

	Level	Technical Performance	Schedule	Cost
Consequence	1	Minimal or no consequence to technical performance	Minimal or no impact	Minimal or no impact
	2	Minor reduction in technical performance or supportability, can be tolerated with little or no impact on program	Able to meet key dates. Slip < 1% of Schedule	Budget increase or unit production cost increases < 1% of Budget
	3	Moderate reduction in technical performance or supportability with limited impact on program objectives	Minor schedule slip. Able to meet key milestones with no schedule float. Slip < 5% of schedule. Sub-system slip > 5% of schedule plus available float	Budget increase or unit production cost increase < 5% of Budget
	4	Significant degradation in technical performance or major shortfall in supportability; may jeopardize program success	Program critical path affected. Slip < 10% of schedule.	Budget increase or unit production cost increase < 10% of Budget
	5	Severe degradation in technical performance; Cannot meet KPP or key technical/supportability threshold; will jeopardize program success	Cannot meet key program milestones. Slip > 10% of schedule	Exceeds APB threshold > 10% of Budget

L. PROBABILITY OF RISK

The team utilized the five levels of probability described in Table 25. The factors that were considered in assigning probability, or likelihood of the risk event occurring, included

- The context of the risk event
- The impacts of the risk if it were realized
- Current project plans
- Whether a mitigation strategy had been defined
- The resources available to execute the mitigation strategy (personnel and time)
- The effectiveness of the strategy at mitigating the occurrence or impact of a risk
- Whether the risk mitigation strategy was being executed

Note: If a mitigation strategy had not been agreed upon, the effects of mitigation were not considered. The probability of occurrence was updated after the mitigation strategy had been selected.

Table 25. DOD Levels of Likelihood Criteria (from Hoeferkamp and Zsak 2007, 25)

	Level	Likelihood	Probability of Occurrence
Likelihood	1	Not Likely	~10%
	2	Low Likelihood	~30%
	3	Likely	~50%
	4	Highly Likely	~70%
	5	Near Certainty	~90%

M. RISK MATRIX

Using the DOD Risk Guide (2006, 11) as the framework, risk rating levels of low, medium and high were assigned based on the combination of impact and probability, as shown in Figure 59.

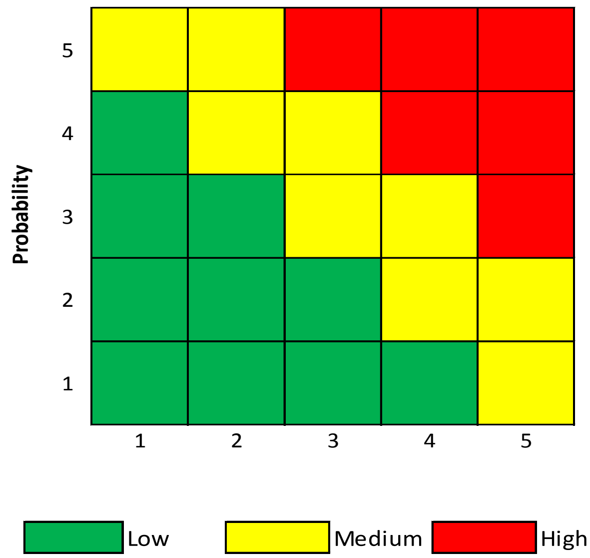


Figure 59. Risk Rating Level Matrix (after DOD, 2006, 11)

N. RISK HANDLING PLANS

According to the DOD Risk Management Guide, responses to risk events generally fall into one of the four categories identified in Table 26: Avoidance, Mitigation, Assumption or Transfer. Mitigation of risks to an acceptable level was generally the preferred method for handling risks, however, each risk handling techniques was evaluated in terms of feasibility, expected effectiveness, and impacts on schedule and performance before the most suitable technique was selected. The team approached each risk according to the strategy identified in Table 26 (DOD 2006).

Table 26. Actions Required Based on Handling Method (from DOD 2013)

Handling Method	Description	Mitigation Plan	Comments
Avoidance	Eliminates the risk, usually by eliminating the root cause	N/A	Close risk. Reduce requirements as a last resort only.
Mitigation	Reduces the risk's probability or expected impact	Required, unless residual risk after mitigation plan selection is LOW.	Monitor risk and track mitigation plan execution
Assumption	Accepts risk	Required, unless waived by PM	Rationale documented under mitigation. Monitor risk for changes to probability or impact
Transfer	Reallocates the risk from one part of a system to another or one organization or functional area to another to reduce overall Project risk	N/A	Close risk. Note that the risk has been transferred in the status and identify the new risk number. Identify the new risk after the transfer and return to risk identification step.

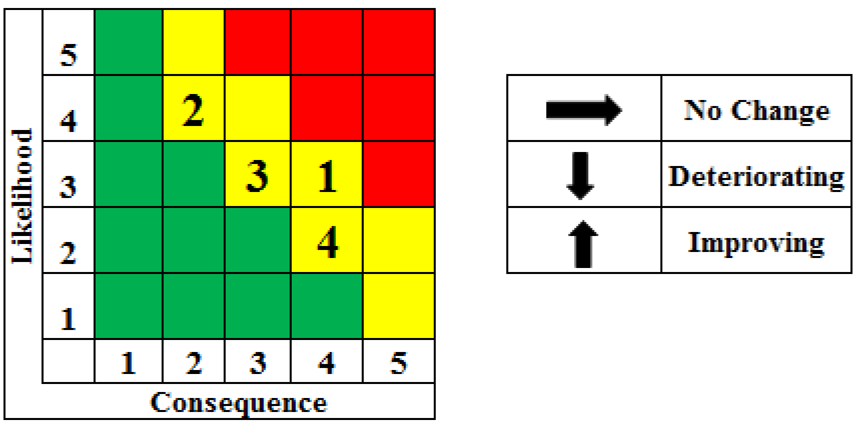
Level	Actions
HIGH	Requires high priority management attention and elevation to higher management levels.
MEDIUM	Requires management attention and may be elevated to higher management levels. Additional contractor emphasis and Government monitoring should be able to overcome difficulties encountered.
LOW	Requires minimal management attention. Monitor risks for changes to probability or impact. Normal contractor effort and Government monitoring should overcome difficulties encountered.

Figure 60. Project Team Actions Required Based on Risk Ratings (from DOD 2013)

O. RISK MITIGATION APPROACH

The risk mitigation approach for the project included monitoring scope, bi-weekly risk briefings, adding necessary personal where required, and receiving guidance from our NPS advisors.

In concert with the DOD Risk Management Guide (2006, 14), the risk mitigation approach defines the steps to take prior to the risk occurring to reduce or eliminate the risk's probability and/or impact. The mitigation approach should focus on the root causes to reduce their probability of occurrence and impact to the project and/or system.



Risk ID	Trend	Title	Description
1	↑	Scope Creep	Delays due to changes in project scope may cause a slip in milestone deliveries
2	→	Key Personal	Work conflicts and TDYs may impact key personal ability to complete assignments within scheduled timeline
3	→	Data Availability	Difficulties in obtaining key data may delay or limit the ability to accurately model the system
4	→	Limited Alternatives	Limitation in viable distribution alternatives may result in product scope changes

Figure 61. Initial Tailored Risk and Key Charts (after Risk Management Guide for DOD Acquisition 2006, 14)

The project risk matrix contains vertical rows that represent the likelihood or probably of occurrence. The rows are ranked from one (lowest probability of occurrence) to five (highest probability of occurrence). The horizontal columns rank the consequence if the risk occurs. The columns are ranked from one (lowest consequence) to five (highest consequence).

The matrix is colored-coded with green for low risk, yellow for medium risk and red for high risk. For example, Scope Creep is the first risk listed (Risk ID 1) in Figure 61. Based on team analysis, the likelihood of occurrence was ranked at three and its consequence a four.

The key describes the projected change of the risk level with “No Change” as an arrow pointing to the right, “Deteriorating” as an arrow pointing down, and “Improving” as an arrow pointing up. The description table contains the risk ID number by order of risk level, its trend {“No Change,” “Deteriorating” or “Improving”), a risk title, and a brief description of the issue.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B. PROTOTYPE DEFINITION

One of the research questions explored by capstone team was "How is prototyping defined with respect to DOD acquisition?" This research led the team to investigate the definition of prototyping outside of the DOD and an attempt to clearly establish what constitutes a prototype and prototyping.

After researching and analyzing many different definitions for these terms, the definitions provided by Samuel Borowski during a 2012 DAU Symposium captured the breadth and depth of the two terms most completely:

- A "prototype" is a test article designed to demonstrate areas of high technical risk that are essential to system success. A prototype need not be a full system, but, in scope and scale, it is tailored to accommodate a series of decisions, and as such, can represent a concept, subsystem, or end item according to the decisions to be made. Rather than reflect the final design, prototypes are built with the expectation that, as decisions are made, change will follow (Borowski 2012).
- "Prototyping" is the practice of testing prototypes, of appropriate scope and scale, for the purpose of obtaining knowledge about some requirement, capability, or design approach. The knowledge obtained informs a decision-making, the output of which results in some degree of change. The degree of allowable change is bounded, in inverse proportion, by the scope and scale of the prototype (Borowski 2012).

Table 27 contains examples of different prototype definitions found during this research. Throughout these definitions, words such as "representation," "demonstration," "model," and "test" appeared frequently. The definitions by Borowski covered all of these areas in one statement.

First, Borowski clearly stated prototypes are test articles that are used in prototyping for obtaining knowledge and capability. Second, the definitions indicate prototypes represent an end item, but are really models for demonstration. The natural tendency with successful prototypes is to push them into production and fielding (Plato

1995). The most successful prototype is still only a test article used to gain knowledge and obtaining another increment of capability and knowledge. While many authors and sources provided insightful definitions of prototyping, the team's research found that Borowski's definitions of prototype and prototyping clearly conveyed the important concepts necessary to understand prototyping.

Table 27. Prototype Definitions

Definition
We define prototype as any representation of a design idea, regardless of medium. This includes a preexisting object when used to answer a design question. We define *designer* as anyone who creates a prototype in order to design, regardless of job title (Hill 1997).
Rapid Prototyping is an agile system, putting solutions for warfighting and intelligence quickly into the hands of users and operators, providing a good portion of needed capabilities upfront in the short term, and gradually upgrading functionality that is based upon continuous input from the field. The integration of standard off-the-shelf commercial technology often makes form fit a minor step in the process of creating vital solutions (Wilbur and Steinhardt 2012).
A production representative article (Gordon 2008).
A "prototype" is a test article designed to demonstrate areas of high technical risk that are essential to system success. A prototype need not be a full system, but, in scope and scale, it is tailored to accommodate series of decisions, and as such, can represent a concept, subsystem, or end item according to the decisions to be made. Rather than reflect the final design, prototypes are built with the expectation that, as decisions are made, change will follow (Borowski 2012).
"Prototyping" is the practice of testing prototypes, of appropriate scope and scale, for the purpose of obtaining knowledge about some requirement, capability, or design approach. The knowledge obtained informs a decision-making process the output of which results in some degree of change. The degree of allowable change is bounded, in inverse proportion, by the scope and scale of the prototype (Borowski 2012).
The word "prototype" was discovered in the interviews to have two possible meanings. In one case, such as in the process that DARPA typically uses or in the one the ORS program has used, a prototype is something developed quickly in the lab, tested in the lab, and used in warfighter operations. This is slightly different than a prototype specifically intended to be used in an operational environment, i.e., as a planned test path for a program of record. An example of the latter would be a fly-off of a new fighter aircraft prototype. The lane as defined herein is meant to consider those rapid prototypes that come out of a rapid environment and are not necessarily intended to become part of a program of record, at least not at the time that the prototypes are tested (Facktor and Colombi 2012).

Definition
A system development methodology based on building and using a model of a system for designing, implementing, testing and installing the system (Tripp and Bichelmeyer 1990).
Prototyping is a quick way to incorporate direct feedback from (real) users into a design. A prototype can be created for the purpose of how it will look, how it will feel, how it will function, where to get it made, and how to make sure it will turn out the way one wants it (Liou 2008).
Prototypes are supposed to be a specification (a living specification, in fact) of the users requirements, not a proposed solution to these needs. The prototype only becomes a solution when the fully specified requirements become equal to a fully specified solution (Carter 1992).
For example, prototyping is an important tool to demonstrate the art of the possible, to expand the realm of the possible, to learn by doing, to free up enormous creativity in government, industry, and academia, to "uncover truth;" and "a concerted effort to mature…, stabilize…, and define/quantify…" (Haller 2013).
The original or model on which something is based or formed. (Dictionary.com 2014)
An original or first model of something from which other forms are copied or developed. A first or early example that is used as a model for what comes later (Merriam-Webster 2014).
A first, typical, or preliminary model of something, especially a machine, from which other forms are developed or copied (Oxford University Press 2014).
A prototype is a product (hardware and/or software) that allows hands on testing in a realistic environment. In scope and scale, it represents a concept, subsystem, or production article with potential utility. It is built in the expectation of change, and is oriented toward generating information improving technical and programmatic decision making. It has purposes and specific objectives other than simply demonstrating that the article meets development contract specifications. The results of prototype testing are used in subsequent decisions, prior to the production decision, influencing system design and requirements formulation, operational utility, and cost and schedule estimates (Drezner 1992).

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX C. DEFINITIONS

This appendix provides the key terms utilized throughout the capstone report. The terms are identified along with definitions of those terms. The terms are defined through either direct quotation or summary of the reference material.

Term	**Direct Quote or Modified Definition**	**Reference**
Action Officer	The Army and sister services use the term, action officer to refer to a staff member (staffer). Action officers shape information and submit recommendations to senior decision makers, that when approved become decisions.	(TRADOC n.d.)
Acquisition Environment	Acquisition programs are structured in phases separated by milestone decisions in accordance with the Life-Cycle Management System established in DOD Instruction 5000.02. In each phase, from defining user needs to disposal, there are important sustainment issues and actions to address.	(DOD 2013)
Analysis of Alternatives (AoA)	The Analysis of Alternatives (AoA) is a documented evaluation of the performance, operational effectiveness, operational suitability, and estimated costs of alternative systems to meet a capability need that has been identified through the Joint Capabilities Integration and Development Systems (JCIDS) process. The AoA assesses the advantages and disadvantages of various materiel alternatives being considered to satisfy the capability need. The AoA also considers the sensitivity of each alternative to possible changes to key	(Morrow 2011)

Term	Direct Quote or Modified Definition	Reference
	assumptions or variables. The AoA is a key input to the process of defining the system capabilities set forth and further refined in the Capability Development Document (CDD).	
Assistant Secretary of Defense for Research and Engineering (ASD[R&E])	The ASD(R&E), under the authority, direction, and control of the USD(AT&L), shall: a. Provide leadership for the DOD on scientific and engineering integrity. b. Facilitate sharing best practices that promote the integrity of DOD scientific and engineering activities. c. Develop clear and specific DOD-wide definitions for the terms "scientific and technical advice," "scientific assessment," "scientific information," "scientific integrity," and "scientific product" as they pertain to scientific and technical advisory committees.	(USD [AT&L] 2012)
Capabilities	The ability to execute a specified course of action. (A capability may or may not be accompanied by an intention.). (JP 1–02)	(JROC 2012)
Capability	A capability that is required to	(JROC 2012)

Term	Direct Quote or Modified Definition	Reference
Requirement (or Requirement)	meet an organization's roles, functions, and missions in current or future operations. To the greatest extent possible, capability requirements are described in relation to tasks, standards, and conditions in accordance with the Universal Joint Task List or equivalent DOD Component Task List. If a capability requirement is not satisfied by a capability solution, then there is also an associated capability gap that carries a certain amount of risk until eliminated. A requirement is considered to be "draft" or "proposed" until validated by the appropriate authority.	
Certification	1.) In the context of the Joint Capabilities Integration and Development System (JCIDS) process, a statement of adequacy by a responsible agency for a specific area of concern in support of the validation process. 2.) A statement by the Milestone Decision Authority (MDA) that certain statutory requirements have been met at Milestone A (Title 10 U.S.C. § 2366a) and at Milestone B (Title 10 U.S.C. § 2366b). 3.) The process within the Office of the Secretary of Defense (OSD) for cooperative Research and Development (R&D) projects authorized under Title 10 U.S.C. § 2350a, whereby candidate projects are screened and those meeting the selection criteria are certified (approved) for implementation pending Memorandum of	(DAU 2012)

Term	Direct Quote or Modified Definition	Reference
	Understanding negotiation and signature and release of funds. Program Elements for these funds are controlled at the OSD and Component headquarters staff levels.	
Conceptual System Design	The Conceptual Design phase is defined by the following actions: Needs Analysis and Identification. Feasibility Analysis. System Operational Requirements. The Maintenance and Support Concept. Technical Performance Measures (TPMs).	(Blanchard and Fabrycky 2011)
Configuration Manager	A discipline applying technical and administrative direction and surveillance to: (1) identify and document the functional and physical characteristics of a configuration item; (2) control changes to those characteristics; and (3) record and report changes to processing and implementation status. Source: JP 6–0	(DOD 2013)
Critical Design Review (CDR)	The Critical Design Review confirms the system design is stable and is expected to meet system performance requirements, confirms the system is on track to achieve affordability and should cost goals as evidenced by the	(DOD 2013)

Term	Direct Quote or Modified Definition	Reference
	detailed design documentation, and establishes the system's initial product baseline.	
Defense Advanced Research Projects Agency (DARPA)	The central research and development organization for the DOD. It manages and directs selected basic and applied research and development projects for DOD, and pursues research and technology where risk and payoff are both very high and where success may provide dramatic advances for traditional military roles and missions.	(DOD 2013)
Deputy Project Manager (DPM)	Deputy to the Project Manager (DPM) or second in command See Project Manager.	
Design	A plan or drawing produced to show the look and function or workings of a building, garment, or other object before it is built or made.	(The Free Dictionary 2014)
DOD Service Research Labs	a : a place equipped for experimental study in a science or for testing and analysis; broadly : a place providing opportunity for experimentation, observation, or practice in a field of study. b : a place like a laboratory for testing, experimentation, or practice. Example Air Force Research Laboratory (AFRL), Army Research Laboratory (ARL), Naval Research Laboratory (NRL)	(Merriam-Webster 2014)
Editor-in-Chief	A person whose job is to be in charge of a group of editors.	(Merriam-Webster 2014)
Entry and Exit Criteria	Entrance criteria are the minimum accomplishments required to be completed by each program prior to entry into the next acquisition	(AcqNotes.com 2014)

Term	Direct Quote or Modified Definition	Reference
	phase or effort. Exit criteria are program-specific accomplishments that must be satisfactorily demonstrated before a program can progress further in the current acquisition phase or transition to the next acquisition phase.	
Enhanced Functional Flow Block Diagrams (EFFBD)	EFFBDs provide data flow overlay to capture data dependencies. EFFBDs represent: (1) functions, (2) control flows, and (3) data flows. An EFFBD specification of a system is complete enough that it is executable as a discrete event model, capable of dynamic, as well as static, validation. EFFBDs provide freedom to use either control constructs or data triggers or both to specify execution conditions for the system functions.	(NASA 2007)
Evaluation Measures (EM)	The specific measure of how well an alternative meets a particular bottom level objective. Scale to measure the degree that we attain an objective (Probability of kill) Evaluation measures are also known as Criteria, Performance Measure, Measure of Effectiveness, Measure of Merit, or Metrics.	(Kirkwood 1997)
Evaluation Measures (EM) Direct	Direct evaluation measures focus specifically on the attainment of the objective in question. Profit is a good example of a direct measure if the objective is to maximize profit.	(Kirkwood 1997)
Evaluation Measures (EM) Proxy	a proxy evaluation measure is one that may focus on the attainment of an associated objective as a surrogate for the actual objective in question. An example of this type of evaluation measure is the use of	(Kirkwood 1997)

Term	Direct Quote or Modified Definition	Reference
	Gross National Product as a measure of economic well-being or the number of tanks destroyed as a measure of success in battle. Generally, direct measures are preferred over proxy measures.	
Evaluation Measures (EM) Natural	A natural evaluation measure is one that is in general use and has a common interpretation by all. In this case, profit is also a good example of a natural measure because it is a commonly accepted financial measure that is calculated in a common way.	(Kirkwood 1997)
Evaluation Measures (EM) Constructed	A constructed evaluation measure is one that is developed for a particular objective. An example of a constructed evaluation measure is level of security classification (i.e., classified, secret, top-secret, etc.). In this case, an evaluation measure was constructed to classify the sensitivity of documents or information because there is no natural measure available. Typically, we prefer to use natural measures rather than those that are constructed.	(Kirkwood 1997)
Fielding	Deploy/Deployment Fielding a weapon system by placing it into operational use with units in the field/fleet.	(DAU 2012)
Functional Analysis	The process of identifying, describing, and relating the functions a system must perform to fulfill its goals and objectives.	(NASA 2007)
Functional Architecture	Provides the foundation for defining the system architecture through the allocation of functions and sub-functions to hardware/software, databases, facilities, and human operations to achieve its mission.	(DOD 2013)

Term	Direct Quote or Modified Definition	Reference
Functional Decomposition	A sub-function under logical decomposition and design solution definition, it is the examination of a function to identify sub-functions necessary for the accomplishment of that function and functional relationships and interfaces.	(NASA 2007)
Full-Scale Development	Having the exact size or proportions of the original: a full-scale replica. Using all possible means, facilities, etc.; complete.	(The Free Dictionary 2014)
Functional Requirement	Serves to translate operational needs into system capabilities. This is the first stage in a sequence of decompositions leading to design. The mission should be examined and characterized in measurable requirement categories such as: quantity, quality, coverage, timeliness, and availability.	(Department of the Navy 2004)
Integrated Definition 0 (IDEF0) Diagrams	IDEF0 is a method designed to model the decisions, actions, and activities of an organization or system. IDEF0 was derived from a well-established graphical language, the Structured Analysis and Design Technique (SADT). The United States Air Force commissioned the developers of SADT to develop a function modeling method for analyzing and communicating the functional perspective of a system. Effective IDEF0 models help to organize the analysis of a system and to promote good communication between the analyst and the customer. IDEF0 is useful in establishing the scope of an analysis, especially for a functional analysis. As a communication tool,	(Knowledge Based Systems 1993)

Term	Direct Quote or Modified Definition	Reference
	IDEF0 enhances domain expert involvement and consensus decision-making through simplified graphical devices. As an analysis tool, IDEF0 assists the modeler in identifying what functions are performed, what is needed to perform those functions, what the current system does right, and what the current system does wrong. Thus, IDEF0 models are often created as one of the first tasks of a system development effort.	
IDEF0 Controls	Specifies the conditions required for the function to produce correct outputs.	(Knowledge Based Systems 1993)
IDEF0 Function	Function name shall be an active verb or phase (process parts, design system, …).	(Knowledge Based Systems 1993)
IDEF0 Inputs	Something that is transformed or consumed by the function.	(Knowledge Based Systems 1993)
IDEF0 Outputs	Data or objects produced by the function.	(Knowledge Based Systems 1993)
IDEF0 Mechanisms	Means that support the execution of the function.	(Knowledge Based Systems 1993)
Integrated Product Teams (IPTs)	Team composed of representatives from appropriate functional disciplines working together to build successful programs, identify and resolve issues, and make sound and timely recommendations to facilitate decision-making. There are three types of IPTs: Overarching IPTs that focus on strategic guidance, program assessment, and issue resolution; Working level IPTs that identify and resolve program issues, determine program status, and seek opportunities for acquisition reform; and Program-level IPTs that focus on program execution	(DAU 2012)

Term	Direct Quote or Modified Definition	Reference
	and may include representatives from both government and industry after contract award.	
Major Defense Acquisition Program (MDAP)	An acquisition program designated by the Under Secretary of Defense for Acquisition, Technology and Logistics (USD [AT&L]) as an MDAP; or estimated to require an eventual total expenditure for Research, Development, Test, and Evaluation (RDT&E), including all planned increments, of more than $365 million in Fiscal Year (FY) 2000 constant dollars or, for procurement, including all planned increments, of more than $2.19 billion in FY 2000 constant dollars.	(DAU 2012)
Materiel Development Decision (MDD)	A review that is the formal entry point into the acquisition process and is mandatory for all programs. A successful MDD may approve entry into the acquisition management system at any point consistent with phase-specific entrance criteria and statutory requirements but will normally be followed by a Materiel Solution Analysis (MSA) phase. The principal documents at this decision point are the Initial Capabilities Document (ICD) and Analysis of Alternatives (AoA) Study Guidance and Plan. A successful MDD normally does not mean that a new acquisition program has been initiated. (DODI 5000.02)	(DAU 2012)
Maturation Planning	The process to plan for the maturing technology and assessing the risk to specific new or novel technologies to meet threshold requirements in development, production, or operation.	(DOD 2013)

Term	Direct Quote or Modified Definition	Reference
Measurable Parameters	Direct measurements that exist to monitor system performance against clearly defined, objective thresholds.	(NASA 2007)
Metrics	Parameters or measures of quantitative assessment used for measurement, comparison or to track performance or production.	(DAU 2012)
Milestone (MS)	The point at which a recommendation is made and approval sought regarding starting or continuing an acquisition program, i.e., proceeding to the next phase. Milestones established by DODI 5000.02 are: Milestone A that approves entry into the Technology Development (TD) phase; Milestone B that approves entry into the Engineering and Manufacturing Development (EMD) phase; and Milestone C that approves entry into the Production and Deployment (P&D) phase. In the context of scheduling, a specific definable accomplishment in the contract network that is recognizable at a particular point in time. Milestones have zero duration, do not consume resources, and have defined entry and exit criteria.	(DAU 2012)
Milestone A	SEE Milestone (MS)	
Milestone B	SEE Milestone (MS)	
Milestone C	SEE Milestone (MS)	
Milestone Decision Authority (MDA)	Designated individual with overall responsibility for a program. The MDA shall have the authority to approve entry of an acquisition program into the next phase of the acquisition process and shall be accountable for cost, schedule, and performance reporting to higher authority, including congressional	(DAU 2012)

Term	Direct Quote or Modified Definition	Reference
	reporting. (DODD 5000.01)	
Mission	(DOD) 1. The task, together with the purpose, that clearly indicates the action to be taken and the reason therefore. Source: JP 3–0 (DOD) 2. In common usage, especially when applied to lower military units, a duty assigned to an individual or unit; a task. Source: JP 3–0 (DOD) 3. The dispatching of one or more aircraft to accomplish one particular task. Source: JP 3–30	(DOD 2013)
Model Simulation Phase	The process of conducting experiments with a model for understanding the behavior of the system modeled under selected conditions or of evaluating various strategies for the operation of the system within the limits imposed by developmental or operational criteria. Simulation may include the use of analog or digital devices, laboratory models, or “test bed” sites. Simulations are usually programmed for solution on a computer; however, in the broadest sense, military exercises and war games are also simulations.	(DOD 2013)
Operational Concept	Organizational/unit structure Basing and deployment description (peacetime, contingency, and wartime) System sustainment concept System logistics concept Maintenance concept Supply management concept Transportation concept Software maintenance concept	(DOD 2013)

Term	Direct Quote or Modified Definition	Reference
	System training concept	
Operational Test & Evaluation (OT&E)	The field test, under realistic conditions, of any item (or key component) of weapons, equipment, or munitions for the purpose of determining the effectiveness and suitability of the weapons, equipment, or munitions for use in combat by typical military users, and the evaluation of the results of such tests.	(DAU 2012)
Operational Requirements	User- or user-representative-generated validated needs developed to address mission area deficiencies, evolving threats, emerging technologies, or weapon system cost improvements. Operational performance requirements from the Capability Development Document (CDD) and Capability Production Document (CPD) form the foundation for weapon system technical specifications and contract requirements.	(DAU 2012)
Pre-Concept Refinement Phase	During this phase, the basic principles of a particular technology are observed, reported, and refined. This stage of development is primarily concerned with analytical and experimental activities meant to provide a proof of concept. Once the technology has been proven to be technically feasible and physically achievable, it is generally accepted to be at a Technology Readiness Level (TRL) of 3 or 4.	(DOD 2013)
Primitive Need	Initial Problem Statement By "primitive" here we mean that the statement represents opinion based mainly on casual observations, but	(Asimov 1962)

Term	**Direct Quote or Modified Definition**	**Reference**
	unsupported by organized evidence. These opinions are often valuable starting points when they come from people who have had the opportunity and have the ability to make observations and to temper them with judgment. A primitive needs statement suggests a problem that might be thought of as an “alleged need,” presented in simple form and ascribed to various kinds of potential customers.	
Problem Refinement Phase	The method to provide a clear statement of goals and objectives for the project and a framework for making and evaluating decisions going forward. Using the background information collected during research, requirements and stakeholder analysis This analysis refined the stakeholder requirements and identified the initial system level requirements relevant to the problem.	(Buede 2009)
Problem Statement	Documents the results of the analysis of a perceived business problem, capability gap, or opportunity (“business need”) undertaken during the Business Capability Definition phase of the Business Capability Life-cycle (BCL) Acquisition Model. (Directive Type Memorandum 11–009).	(DAU 2012)
Procurement Regulations	Act of buying goods and services for the government and a rule or order issued by an executive authority or regulatory agency of a government and having the force of law.	(DAU 2012)
Project Manager (PM),	The program manager (PM) is the acquisition team leader and is	(DOD 2013)

Term	Direct Quote or Modified Definition	Reference
	responsible for ensuring that the acquisition plan is properly executed and the desired results are achieved. The PM provides coordination and facilitates communication among the acquisition team members, closely tracks the milestone schedule, and provides leadership and guidance to overcome and resolve any problems or delays. This individual is responsible for drafting the PWS, which means ensuring that performance requirements are clearly and concisely defined and articulated. PMs identify, plan, and control various areas, such as delivery requirements, scheduling, market research, COR nomination, cost estimating, budgeting, and specific project formulation. The PM normally participates in the source selection as well. This individual serves as the principal technical expert, is most familiar with the requirement, best able to identify potential technical tradeoffs, and whether the requirement can be met by a commercial solution.	
Prototype	A "prototype" is a test article designed to demonstrate areas of high technical risk that are essential to system success. A prototype need not be a full system, but, in scope and scale, it is tailored to accommodate a series of decisions, and as such, can represent a concept, subsystem, or end item according to the decisions to be made. Rather than reflect the final design, prototypes are built with the expectation that, as	(Borowski, 2012)

Term	Direct Quote or Modified Definition	Reference
	decisions are made, change will follow.	
Prototyping	“Prototyping” is the practice of testing prototypes, of appropriate scope and scale, for the purpose of obtaining knowledge about some requirement, capability, or design approach. The knowledge obtained informs a decision-making, the output that results in some degree of change. The degree of allowable change is bounded, in inverse proportion, by the scope and scale of the prototype.	(Borowski, 2012)
Regulatory	Requirements directed by military regulations.	(Department of the Navy 2004)
Requirements	1.) The need or demand for personnel, equipment, facilities, other resources, or services, by specified quantities for specific periods of time or at a specified time. 2.) For use in budgeting, item requirements should be screened as to individual priority and approved in the light of total available budget resources.	(DAU 2012)
Requirements Analysis	Encompasses the definition and refinement of system, subsystem, and lower-level functional and performance requirements and interfaces to facilitate the Architecture Design process. Establishes the functional architecture that expresses the detailed functional, interface, and temporal aspects of the system to unambiguously communicate system behavior in its intended environment, and the development of lower tier functional and performance requirements that need to be allocated to the system physical architecture.	(DAU 2012)

Term	Direct Quote or Modified Definition	Reference
Reviews	The discrete process of gathering and evaluating information to make a decision about a program. Examples are milestone reviews and other program decision reviews.	(DAU 2012)
Risk	A measure of future uncertainties in achieving program performance goals and objectives within defined cost, schedule, and performance constraints. Risk can be associated with all aspects of a program (e.g., threat, technology, maturity, supplier capability, design maturation, performance against plan) as these aspects relate across the Work Breakdown Structure (WBS) and Integrated Master Schedule (IMS). Risks have three components: 1.) A future root cause (yet to happen), which, if eliminated or corrected, would prevent a potential consequence from occurring; 2.) a probability (or likelihood) assessed at the present time of that future root cause occurring; and 3.) a consequence (or effect) of that future occurrence. (Risk Management Guide for DOD Acquisition, Sixth Edition)	(DAU 2012)
Risk Analysis	The activity that examines each identified risk to refine the description of the risk, isolate the cause, and determine the effects in setting risk mitigation priorities. It considers the likelihood of root cause occurrence; identifies possible consequences in terms of performance, schedule, and cost; and identifies the risk level in terms of high (red), medium (yellow), and low (green) on a	(DAU 2012)

Term	Direct Quote or Modified Definition	Reference
	Risk Reporting Matrix. (Risk Management Guide for DOD Acquisition, Sixth Edition) See Risk Reporting Matrix.	
Risk Assessment	The process of determining, identifying risks and future root causes, developing mitigation Plans.	(DOD 2006)
Risk Management Plan (RMP)	The plan for with risk. It includes planning for risk, assessing (identifying and analyzing) risk issues, developing risk handling options, monitoring risks to determine how risks have changed, and documenting the overall risk management program.	(DOD 2003)
Risk Manager (RM)	Leads the management process to ensure that the risks are prioritized Receive appropriate level of management attention.	(DOD 2006)
Simulation	A method for implementing a model. It is the process of conducting experiments with a model for understanding the behavior of the system modeled under selected conditions or of evaluating various strategies for the operation of the system within the limits imposed by developmental or operational criteria. Simulation may include the use of analog or digital devices, laboratory models, or "testbed" sites. Simulations are usually programmed for solution on a computer; however, in the broadest sense, military exercises and war games are also simulations.	(DAU 2012)
State, Investigate, Model, Integrate, Launch, Assess and Re-evaluate (SIMILAR)	State the problem, Investigate alternatives, Model the system, Integrate, Launch the system, Assess performance, and Re-evaluate. These seven functions	(Gissing 1998)

Term	Direct Quote or Modified Definition	Reference
	can be summarized with the acronym SIMILAR: State, Investigate, Model, Integrate, Launch, Assess, and Reevaluate.	
Stakeholder	A person or organization that is actively involved with the program, or whose interests may be positively or negatively impacted by the program. Stakeholders may include: Program Manager, Program Office, Users, Contractors, Congress, Other programs etc...	(DAU 2012)
Statutory	Required, permitted, or enacted by statute.	(Oxford University Press 2014)
Subject Matter Experts (SME)	A person who is an expert in a particular area or topic. The term domain expert is frequently used in expert systems software development, and there the term always refers to the domain other than the software domain. A domain expert is a person with special knowledge or skills in a particular area of endeavor. An accountant is an expert in the domain of accountancy, for example. The development of accounting software requires knowledge in two different domains, namely accounting and software. Some of the development workers may be experts in one domain and not the other. SME should also have basic knowledge on other technical subjects.	(The Free Dictionary 2014)
Systems	1.) The organization of hardware, software, material, facilities, personnel, data, and services needed to perform a designated function with specified results, such as the gathering of specified data, its processing, and delivery to	(DAU 2012)

Term	Direct Quote or Modified Definition	Reference
	users. 2.) A combination of two or more interrelated pieces of equipment (or sets) arranged in a functional package to perform an operational function or to satisfy a requirement.	
Subsystems	A functional grouping of components that combine to perform a major function within an element such as electrical power, attitude control, and propulsion.	(DAU 2012)
System Characteristics	A system distinguishing trait, quality, or property timing, process behavior, or various performance measures.	(Merriam-Webster 2014)
Systems Engineering (SE)	An interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, and then proceeding with design synthesis and system validation while considering the complete problem: operations, cost and schedule, performance, training and support, test, manufacturing, and disposal. SE considers both the business and the technical needs of all customers with the goal of providing a quality product that meets the user needs,	(INCOSE 2010)
System Level Requirements	At the System Level Requirements are the need or demand for personnel, equipment, facilities, other resources, or services, by specified quantities for specific periods of time or at a specified time. 2.) For use in budgeting, item requirements should be screened as to individual priority and approved in the light of total available budget resources.	(DAU 2012)

Term	Direct Quote or Modified Definition	Reference
System life cycle	An examination of a system or proposed system that addresses all phases of its existence to include system conception, design and development, production and/or construction, distribution, operation, maintenance and support, retirement, phase-out and disposal.	(Blanchard and Fabrycky 2011, 29)
System Modeling	The interdisciplinary use to analysis, design and conceptualize systems specific techniques such as the Functional Flow Block Diagram and IDEF0 sometimes for simulation. Using functional decomposition the models and can be linked to requirements models for further systems partition.	(Blanchard and Fabrycky 2011)
Termination	Cessation; conclusion; end in time or existence. When used in connection with litigation, the term signifies the final determination of the action. The termination or cancellation of a contract signifies the process whereby an end is put to whatever remains to be performed under the contract. It differs from Rescission, which refers to the restoration of the parties to the positions they occupied prior to the contract. The termination of a lease refers to the severance of the Landlord and Tenant relationship before the leasehold term expires through the ordinary passage of time.	(The Free Dictionary 2014)
Technology Development Strategy (TDS)	The TDS should also include the specific new sustainment related technologies required to achieve the Sustainment KPP/KSAs. Specific emphasis should be placed on technologies required to achieve logistics performance (including	(DOD 2013)

Term	Direct Quote or Modified Definition	Reference
	reliability) over what is currently achieved in today's operational environment.	
Technology Readiness Level (TRL)	1. Basic principles observed and reported. Lowest level of technology readiness. Scientific research begins to be translated into applied research and development. Examples might include paper studies of a technology's basic properties. 2. Technology concept and/or application formulated. Invention begins. Once basic principles are observed, practical applications can be invented. Applications are speculative and there may be no proof or detailed analysis to support the assumptions. Examples are limited to analytic studies. 3. Analytical and experimental critical function and/or characteristic proof of concept. Active research and development is initiated. This includes analytical studies and laboratory studies to physically validate analytical predictions of separate elements of the technology. Examples include components that are not yet integrated or representative. 4. Component and/or breadboard validation in laboratory environment. Basic technological components are integrated to establish that they will work together. This is relatively "low fidelity" compared to the eventual system. Examples include integration of "ad hoc" hardware in	(DOD 2013)

Term	Direct Quote or Modified Definition	Reference
	the laboratory. 5. Component and/or breadboard validation in relevant environment. Fidelity of breadboard technology increases significantly. The basic technological components are integrated with reasonably realistic supporting elements so it can be tested in a simulated environment. Examples include “high fidelity” laboratory integration of components. 6. System/subsystem model or prototype demonstration in a relevant environment. Representative model or prototype system, which is well beyond that of TRL 5, is tested in a relevant environment. Represents a major step up in a technology’s demonstrated readiness. Examples include testing a prototype in a high-fidelity laboratory environment or in simulated operational environment. 7. System prototype demonstration in an operational environment. Prototype near, or at, planned operational system. Represents a major step up from TRL 6, requiring demonstration of an actual system prototype in an operational environment such as an aircraft, vehicle, or space. Examples include testing the prototype in a test bed aircraft. 8. Actual system completed and qualified through test and demonstration. Technology has	

Term	Direct Quote or Modified Definition	Reference
	been proven to work in its final form and under expected conditions. In almost all cases, this TRL represents the end of true system development. Examples include developmental test and evaluation of the system in its intended weapon system to determine if it meets design specifications. 9. Actual system proven through successful mission operations. Actual application of the technology in its final form and under mission conditions, such as those encountered in operational test and evaluation. Examples include using the system under operational mission conditions.	
Technology Readiness Assessment (TRA).	A systematic, metrics-based process that assesses the maturity of, and the risk associated with, critical technologies to be used in Major Defense Acquisition Programs (MDAPs). It is conducted by the Program Manager (PM) with the assistance of an independent team of subject matter experts (SMEs). It is provided to the Assistant Secretary of Defense for Research and Engineering (ASD [R&E]) and will provide part of the basis upon which he advises the Milestone Decision Authority (MDA) at Milestone (MS) B or at other events designated by the MDA to assist in the determination of whether the technologies of the program have acceptable levels of risk—based in part on the degree to which they have been	(DOD 2013)

Term	**Direct Quote or Modified Definition**	**Reference**
	demonstrated (including demonstration in a relevant environment)—and to support risk-mitigation plans prepared by the PM.	
Test and Evaluation (T&E)	Process by which a system or components are exercised and results analyzed to provide performance-related information. The information has many uses including risk identification and risk mitigation and empirical data to validate models and simulations. T&E enables an assessment of the attainment of technical performance, specifications, and system maturity to determine whether systems are operationally effective, suitable and survivable for intended use, and/or lethal. There are various types of T&E defined in statute or regulation: Developmental Test and Evaluation (DT&E), Operational Test and Evaluation (OT&E), Live Fire Test and Evaluation (LFT&E), and Interoperability Test and Certification. See Operational Test and Evaluation (OT&E), Initial Operational Test and Evaluation (IOT&E), Developmental Test and Evaluation (DT&E), and Live Fire Test and Evaluation (LFT&E).	(DAU 2012)
Threat	The sum of the potential strengths, capabilities, and strategic objectives of any adversary that can limit or negate U.S. mission accomplishment or reduce force, system, or equipment effectiveness.	(DAU 2012)
Technological Environments	The Environment 1. Relating to or involving technology, especially scientific technology.	(The Free Dictionary 2014)

Term	**Direct Quote or Modified Definition**	**Reference**
	2. Affected by or resulting from scientific and industrial progress.	
Technology	The use of science in industry, engineering, etc., to invent useful things or to solve problems a machine, piece of equipment, method, etc., that is created by technology.	(Merriam-Webster 2014)
Technology Transition	Process of inserting critical technology into military systems to provide an effective weapons and support system in the quantity and quality needed by the warfighter to carry out assigned missions.	(DAU 2012)
Test Article	An item built, constructed, coded, or otherwise implemented, for checking conformance to specified requirements or for checking validation against acquirer requirements for the item.	(Department of the Navy 2004)
Use Case	Describes how a user interacts with the system by defining the steps required to accomplish a specific goal (e.g., burning a list of songs onto a CD). Variations in the sequence of steps describe various scenarios (e.g., what if all the songs in the list don't fit on one CD?).	(Pressman 2010)
User	An operational command or agency that receives or will receive benefit from an acquired system. Combatant Commands (COCOM) and their Component commands are users. There may be more than one user for a system. Because the military services are required to organize, equip, and train forces for the CCMDs, they are also seen as users for systems.	(DAU 2012)
Value System Design	The process of determining the values of critical stakeholders by	(Kirkwood 1997)

Term	Direct Quote or Modified Definition	Reference
	expanding the effective need of our system into critical functions, sub-functions and objectives with evaluation measures.	
Value Model	A mechanism used to evaluate how well each alternative meets the clients' needs with a specific measure of how well an alternative meets a particular bottom level objective is referred to as an evaluation measure.	(Kirkwood 1997)
Weapon System Acquisition Reform Act (WSARA)	WSARA, Public Law 111–23, was enacted in 2009 with the purpose of putting Major Defense Acquisition Programs (MDAPs) on a sound footing from the outset by requiring additional focus on Systems Engineering (SE); management of technology risk; earlier, realistic estimates of program cost; funding to Independent Cost Estimates (ICEs); and renewed emphasis on competition, including competitive prototyping at the system or key subsystem level prior to program initiation.	(DAU 2012)
Weapon System Development	Development of Items that can be used directly by the Armed Forces to carry out combat missions.	(DAU 2012)

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX D. TRACEABILITY REPORT

The traceability report is an output from the Innoslate Model that was produced for this project. This report shows the relationship from the functions with their description to the functional requirements with their descriptions. This is an automated report from the Innoslate software that the team utilized to show the relationship of the functions to the functional requirements.

Entity Title	Entity Description	Traced from Entity Title	Traced from Entity Description
EXT.F.1 Perform Government Entities Functions	This entity serves to encapsulate the external functions that are performed by the External Government Entities Node. This function produces two outputs that are used in our model. This function produces 2 outputs that are used in our model. The first output is Regulations & Policies that are produced by several different entities throughout the Government, some of which include the White House, the Pentagon, and the Congress to name a few. The second output is funding, that through the budgeting process is touched and approved by many different Government Entities as well. The funding once approved is provided to the Project Office, or Customer in the case of our model, to be expended on system development.	REQ.1.3.1 Perform Government Entities Activities	
EXT.F.2 Perform End User Functions	This entity serves to encapsulate the external functions that are performed by the End User Entity Node. This function produces three outputs that are used in our model.	REQ.1.3.2 Perform End User Activities	

Entity Title	Entity Description	Traced from Entity Title	Traced from Entity Description
	This function receives 1 inputs and produces 3 outputs that are used in our model.		
EXT.F.3 Perform Customer Functions	This entity serves to encapsulate the external functions that are performed by the Customer Entity Node. This function receives four inputs and produces one outputs that are used in our model. This function receives 4 inputs (3 triggers and 1 input) and produces 1 output that are used in our model.	REQ.1.3.3 Perform Customer Activities	
F.0 Perform DOD Prototyping System Functions	This element represents the activities performed by the DOD Prototyping System. Child elements represent activities that are within the system boundary of the DOD Prototyping System. This function is composed of six different sub-functions: Assess Feasibility; Produce Technology Development Plan; Mature Technology; Transition Technology; Redefine/Terminate Program; and Technology Maturation Closeout. This function is further decomposed by 6 children functions and receives 3 inputs (2 triggers and 1 input) and generates 2 outputs that are used in our model.	0 DOD Prototype System Requirements	These are the requirements for the system architecture. The system is the solution under development or analysis - everything inside the system boundary (may be a System of Systems). The higher level requirements trace back to the capabilities. Requirements are decomposed from high level solution-neutral capabilities and requirements all the way down to solution-oriented system specifications.
F.1 Assess Feasibility	This activity is an internal function of the system under development. This function serves to assess the overall feasibility of the technology being requested for maturation. This function is further	REQ.1.4.1 Assess Feasibility	The system shall assess project feasibility.

Entity Title	Entity Description	Traced from Entity Title	Traced from Entity Description
	decomposed by 3 children function and receives 3 inputs and generates 2 outputs.		
F.1.1 Technology Readiness Assessment	This activity is an internal sub-function of the system under development. This sub-function requires a Technology Readiness Assessment (TRA) in that the system in directly involved in determining the assessment. This serves to ensure that the system concurs with the TRL of the technology coming into the system. This sub-function receives 3 inputs and generates 2 outputs.	REQ.1.4.1.1 Technology Readiness Assessment	The system shall perform a technology readiness assessment.
F.1.2 Assess Technical Feasibility	This activity is an internal sub-function of the system under development. This sub-function requires an assessment of the technical feasibility of maturing the technology being introduced into the system. As an example: if the technology being requested for maturation is a space elevator then this is not a Technical Feasibility at this point in time. This sub-function receives 1 input and generates 2 outputs.	REQ.1.4.1.2 Assess Technical Feasibility	The system shall assess technology feasibility.
F.1.3 Assess Programmatic Feasibility	This activity is an internal sub-function of the system under development. This sub-function requires an assessment to be performed to determine if the programmatic constraints of the project office is feasible enough to allow for the maturation of the technology within those constraints. This sub-function receives 2 inputs and generates 2 outputs.	REQ.1.4.1.3 Assess Programmatic Feasibility	The system shall assess programmatic feasibility.

Entity Title	Entity Description	Traced from Entity Title	Traced from Entity Description
F.2 Produce Technology Development Plan	This activity is an internal function of the system under development. This function serves to produce a plan that will layout how the technology will be developed and matured. The desired output of this function is a Transition Development Agreement that is signed by the System under development and the Customer (Project Office). This function is further decomposed by 4 children function and receives 3 inputs and generates 2 outputs.	REQ.1.4.2 Produce Technology Development Plan	The system shall produce a technology development plan.
F.2.1 Determine Maturation Risks for Next Phase	This activity is an internal sub-function of the system under development. This sub-function requires an assessment of the current risks that have been defined for the technological development and the identification of anticipated/projected risks for the technological maturation. If the risks are deemed acceptable by all parties involved then the output OI.2.1 is sent otherwise OI.0.4 is sent. This sub-function receives 3 inputs and generates 2 outputs.	REQ.1.4.2.1 Determine Maturation Risks for Next Phase	The system shall determine maturation risks for the next phase.
F.2.2 Determine Maturation Costs for Next Phase	This activity is an internal sub-function of the system under development. This sub-function serves to produce a cost estimate to mature the technology from it's current TRL to the next. This sub-function receives 2 inputs and generates 2 outputs.	REQ.1.4.2.2 Determine Maturation Costs for Next Phase	The system shall determine maturation costs for the next phase.
F.2.3 Determine Maturation Schedule for Next	This activity is an internal sub-function of the system under development. This	REQ.1.4.2.3 Determine Maturation	The system shall determine maturation

Entity Title	Entity Description	Traced from Entity Title	Traced from Entity Description
Phase	sub-function serves to produce a schedule estimate for the maturing of the technology from it's current TRL to the next. This sub-function receives 2 inputs and generates 2 outputs.	Schedule for Next Phase	schedule for the next phase.
F.2.4 Finalize Plan for Agreement	This activity is an internal sub-function of the system under development. This sub-function serves to produce the finalized Plan for the maturing of the technology from it's current TRL to the next, for agreement. This sub-function receives 2 inputs and generates 2 outputs.	REQ.1.4.2.4 Finalize Plan for Agreement	The system shall finalize a technology development plan for agreement.
F.3 Technology Mature	This activity is an internal function of the system under development. This function serves as the heart and soul of the system. It is where the technology maturation actually takes place. The plan that was generated in function OA.2 is used to define how the technology maturation will be executed during this function. The technology to be matured is expected to entering at a TRL 3 or TRL 4 and being matured to a TRL 4 or TRL 5 respectively. This function is further decomposed by 4 children function and receives 3 inputs and generates 4 outputs.	REQ.1.4.3 Mature Technology	The system shall mature technology.
F.3.1 Design Prototypes	Development a plan to build a prototype. This activity is an internal sub-function of the system under development. This sub-function is where the technology begins to get matured starting with a prototype design. This sub-	REQ.1.4.3.1 Design Prototypes	The system shall design prototypes.

Entity Title	Entity Description	Traced from Entity Title	Traced from Entity Description
	function receives 3 inputs and generates 3 outputs.		
F.3.1.1 Define System Boundary	The boundary for the prototype is defined to identify what the system is and what the context around the system is. This activity is an internal sub-sub-function of the system under development. This sub-sub-function serves to define the system boundary of the technology under development. This sub-function receives 3 inputs and generates 3 outputs.	REQ.1.4.3.1.1 Define System Boundary	The system shall define the prototype system boundary.
F.3.1.2 Derive System Threads	This activity is an internal sub-sub-function of the system under development. This sub-sub-function serves to derive and define the system threads for the technology under development. This sub-function receives 2 inputs and generates 3 outputs.	REQ.1.4.3.1.2 Derive System Threads	The system shall derive prototype system threads.
F.3.1.3 Derive Component Hierarchy	This activity is an internal sub-sub-function of the system under development. This sub-sub-function serves to derive and define the component hierarchy for the technology under development. This sub-function receives 2 inputs and generates 3 outputs.	REQ.1.4.3.1.3 Derive Component Hierarchy	The system shall derive prototype component hierarchy.
F.3.1.4 Allocate Behavior to Components	This activity is an internal sub-sub-function of the system under development. This sub-sub-function serves to allocate the technological behaviors to the components that were defined under the previous function. This sub-function receives 2 inputs and	REQ.1.4.3.1.4 Allocate Behavior to Components	The system shall allocate behavior to prototype components.

Entity Title	Entity Description	Traced from Entity Title	Traced from Entity Description
	generates 3 outputs.		
F.3.1.5 Perform Modeling and Simulations	This activity is an internal sub-sub-function of the system under development. This sub-sub-function serves to perform and execute modeling and simulations on the different designs that has been generated by the preceding functions. This sub-function receives 2 inputs and generates 3 outputs.	REQ.1.4.3.1.5 Perform Modeling	The system shall perform modeling.
		REQ.1.4.3.1.6 Perform Simulations	The system shall perform simulations.
F.3.1.6 Perform Effectiveness & Feasibility Analysis	This activity is an internal sub-sub-function of the system under development. This sub-sub-function serves to evaluate the results produce by the Modeling and Simulations function that were performed by the previous function, OA.3.1.5. This sub-function receives 2 inputs and generates 3 outputs.	REQ.1.4.3.1.7 Perform Effectiveness Analysis	The system shall perform effectiveness analysis.
		REQ.1.4.3.1.8 Perform Feasibility Analysis	The system shall perform feasibility analysis.
F.3.1.7 Select Design	This activity is an internal sub-sub-function of the system under development. This sub-sub-function serves select the best design for the capability/technology to move forward with based on the results of the analysis from the models and sims data. This sub-function receives 2 inputs and generates 3 outputs.	REQ.1.4.3.1.9 Select Design	The system shall select a prototype design.
F.3.1.8 Define Resources, Error Detection, & Recovery	This activity is an internal sub-sub-function of the system under development. This sub-sub-function serves to define the additional layers of technological concern for the development such as resources, error detection and recovery	REQ.1.4.3.1.10 Define Resources	The system shall define prototype resources.
		REQ.1.4.3.1.11 Define Error Detection	The system shall define error detection.
		REQ.1.4.3.1.12 Define Recovery	The system shall define recovery.

Entity Title	Entity Description	Traced from Entity Title	Traced from Entity Description
	process/procedures. This sub-function receives 2 inputs and generates 3 outputs.		
F.3.1.9 Generate Documentation & Specifications	This activity is an internal sub-sub-function of the system under development. This sub-sub-function serves to ensure that the design documentation and specifications have been generated and are in the proper format. This sub-function receives 3 inputs and generates 3 outputs.	REQ.1.4.3.1.13 Generate Documentation	The system shall generate documentation.
		REQ.1.4.3.1.14 Generate Specifications	The system shall generate specifications.
F.3.2 Build Prototypes	Creation, integration, and assembly of prototype hardware and software. This activity is an internal sub-function of the system under development. This sub-function is responsible for building the prototype in accordance with the design that was generated by sub-function OA.3.1. This sub-function receives 3 inputs and generates 4 outputs.	REQ.1.4.3.2 Build Prototypes	The system shall build prototypes.
F.3.2.1 Build Prototype Hardware	This activity is an internal sub-sub-function of the system under development. This sub-sub-function serves to build/produce a hardware prototype based on the design selected in function OA.3.1. This sub-function receives 3 inputs and generates 3 outputs.	REQ.1.4.3.2.1 Build Prototype Hardware	The system shall build prototype hardware.
F.3.2.2 Build Prototype Software	This activity is an internal sub-sub-function of the system under development. This sub-sub-function serves to build/produce a software prototype based on the design selected in function OA.3.1. This sub-function	REQ.1.4.3.2.2 Build Prototype Software	The system shall build prototype software.

Entity Title	Entity Description	Traced from Entity Title	Traced from Entity Description
	receives 3 inputs and generates 3 outputs.		
F.3.2.3 Integrate Prototype Components	This activity is an internal sub-sub-function of the system under development. This sub-sub-function serves to integrate the hardware and software prototypes produced by sub-sub-functions OA.3.2.1 and OA.3.2.2. This sub-function receives 4 inputs and generates 3 outputs.	REQ.1.4.3.2.3 Integrate Prototype Components	The system shall integrate prototype components.
F.3.2.4 Perform Component Integration Testing	This activity is an internal sub-sub-function of the system under development. This sub-sub-function serves to verify and validate the integration of the prototypes via testing against the requirements and design. This sub-function receives 3 inputs and generates 4 outputs.	REQ.1.4.3.2.4 Perform Component Integration Testing	The system shall perform component integration testing.
F.3.3 Demonstrate Prototype in Simulated Environment	Creation of a relevant controlled environment to demonstrate a prototype. This activity is an internal sub-function of the system under development. This sub-function serves to perform the demonstration of the prototype in a simulated environment. This sub-function receives 3 inputs and generates 4 outputs.	REQ.1.4.3.3 Demonstrate Prototype in Simulated Environment	The system shall demonstrate the prototype in a simulated environment
F.3.3.1 Model Simulated Environment	This activity is an internal sub-sub-function of the system under development. This sub-sub-function serves to build/produce an approved simulated environment for prototype testing. This sub-function receives 3 inputs and generates 3 outputs.	REQ.1.4.3.3.1 Model Simulated Environment	The system shall model a simulated environment.

Entity Title	Entity Description	Traced from Entity Title	Traced from Entity Description
F.3.3.2 Run Prototype in Simulated Environment	This activity is an internal sub-sub-function of the system under development. This sub-sub-function is responsible for executing the prototype within the simulated environment and collect data on how the technology under development reacts and responds. This sub-function receives 2 inputs and generates 3 outputs.	REQ.1.4.3.3.2 Run Prototype in Simulated Environment	The system shall run the prototype in a simulated environment.
F.3.3.3 Evaluate Results	This activity is an internal sub-sub-function of the system under development. This sub-sub-function serves to evaluate the results and data that were produced by the prototype under test within the simulated environment. This sub-function receives 2 inputs and generates 4 outputs.	REQ.1.4.3.3.3 Evaluate Results	The system shall evaluate the results of the simulation.
F.3.4 Demonstrate Prototype in Operational Environment	Creation of a relevant operational environment to demonstrate a prototype. This activity is an internal sub-function of the system under development. This sub-function serves to perform the demonstration of the prototype in an Operational environment. This sub-function receives 4 inputs and generates 3 outputs.	REQ.1.4.3.4 Demonstrate Prototype in Operational Environment	The system shall demonstrate the prototype in an operational environment.
F.3.4.1 Validate Operational Environment	This activity is an internal sub-sub-function of the system under development. This sub-sub-function serves to validate the operational environment that the technology under development will be tested in. This sub-function receives 3 inputs and generates 3 outputs.	REQ.1.4.3.4.1 Validate Operational Environment	The system shall validate the operational environment.

Entity Title	Entity Description	Traced from Entity Title	Traced from Entity Description
F.3.4.2 Demonstrate Prototype in Operational Environment	This activity is an internal sub-sub-function of the system under development. This sub-sub-function is responsible for executing the prototype within the operational environment and collect data on how the technology under development reacts and responds. This sub-function receives 3 inputs and generates 3 outputs.	REQ.1.4.3.4.2 Demonstrate Prototype in Operational Environment	The system shall demonstrate the prototype in an operational environment.
F.3.4.3 Evaluate Results	This activity is an internal sub-sub-function of the system under development. This sub-sub-function serves to evaluate the results and data that were produced by the prototype under test within the operational environment. This sub-function receives 2 inputs and generates 3 outputs.	REQ.1.4.3.4.3 Evaluate Results	The system shall evaluate the demonstration results.
F.4 Transition Technology	This activity is an internal function of the system under development. This function serves to determine whether the technology under development should be transitioned out of the DOD Prototyping System and into the next phase of Acquisition Development. This function is further decomposed by 3 children function and receives 2 inputs and generates 2 outputs.	REQ.1.4.4 Transition Technology	The system shall transition technology.
F.4.1 Finalize Technology Transition Artifacts	This activity is an internal sub-function of the system under development. This sub-function serves to finalize the artifacts produced during the development in order to support the Transition Technology Readiness	REQ.1.4.4.1 Finalize Technology Transition Artifacts	The system shall finalize technology transition artifacts.

Entity Title	Entity Description	Traced from Entity Title	Traced from Entity Description
	Assessment (TRA). This sub-function receives 2 inputs and generates 1 output.		
F.4.2 Perform Technology Readiness Assessment for Transition	This activity is an internal sub-function of the system under development. This sub-function requires a Technology Readiness Assessment (TRA) in that the system in directly involved in determining the assessment. This serves to ensure that the system concurs with the TRL of the technology coming into the system. This sub-function receives 3 inputs and generates 2 outputs.	REQ.1.4.4.2 Perform Technology Readiness Assessment for Transition	The system shall perform a technology readiness assessment for transition.
F.4.3 Transition Technology Artifacts	Will show the progress up to TRL level 6, the requirements that have been met along with the strategy, to continue development of the technology. This activity is an internal sub-function of the system under development. This sub-function requires a Technology Readiness Assessment (TRA) in which the system in directly involved in determining the assessment. This serves to ensure that the system concurs with the TRL of the technology coming into the system. This sub-function receives 3 inputs and generates 2 outputs.	REQ.1.4.4.3 Transition Technology Artifacts	The system shall transition technology artifacts.
F.5 Redefine/Terminate Program	This activity is an internal function of the system under development. This function serves to determine whether the plan for the technology under development should be redefined or if the Program should be	REQ.1.4.5 Redefine/Terminate Program	The system redefine or terminate the program.

Entity Title	Entity Description	Traced from Entity Title	Traced from Entity Description
	Terminated instead. This function is further decomposed by 3 children function and receives 4 inputs and generates 2 outputs.		
F.5.1 Program Determination	This activity is an internal sub-function of the system under development. This sub-function serves as an opportunity for the state of the program to be reassessed to determine whether the program should be terminated or re-planned. This sub-function receives 4 inputs and generates 3 outputs.	REQ.1.4.5.1 Program Determination	The system shall make a program determination.
F.5.2 Redefine Program Plan	This activity is an internal sub-function of the system under development. This sub-function serves to notify the system that the program will be redefined. This sub-function receives 1 input and generates 1 output.	REQ.1.4.5.2 Redefine Program Plan	The system shall redefine the technology development plan.
F.5.3 Capture Issue Metrics	This activity is an internal sub-function of the system under development. This sub-function serves to determine and collect issue metrics for faults or failures within the system. This supports the ability for the system to identify areas for process improvements. This sub-function receives 1 inputs and generates 1 outputs.	REQ.1.4.5.3 Capture Issue Metrics	The system shall capture issue metrics.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX E. SIMULATION RESULTS EXPLANATION

The figures included in this appendix provide an explanation of the model simulation results. The result of conducting the simulation in Innoslate is a Gant chart illustrating the actual result of the simulation on a timeline describing when each function was utilized. The figures in this section provide examples to explain the components of the Gant chart and how to read the characters displayed on the timeline. The examples provide a context to understand the simulation results.

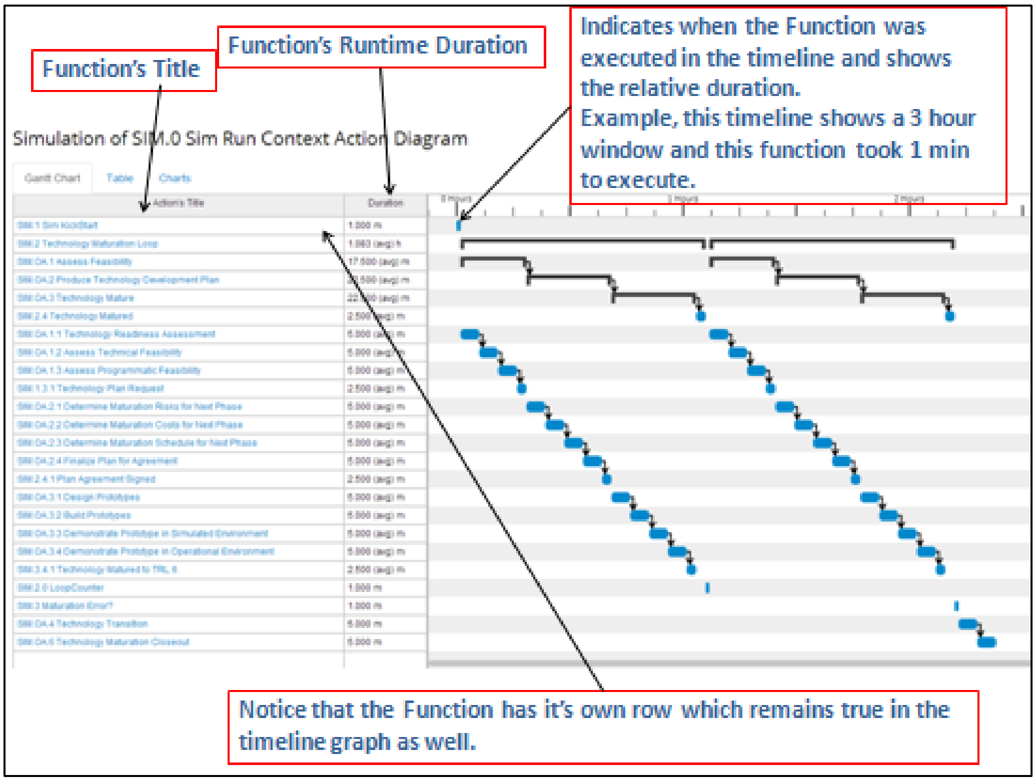


Figure 62. Simulation Gant Chart Function Explanation

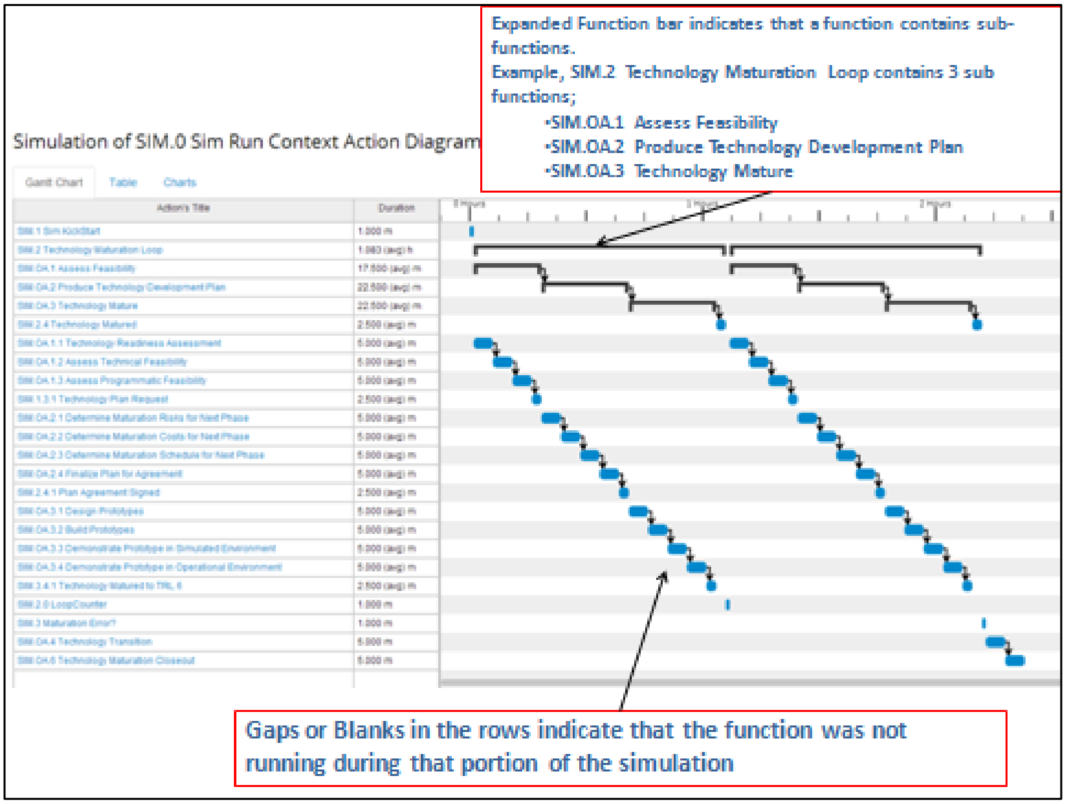


Figure 63. Simulation Gant Chart Function and Sub-function Timeline Explanation

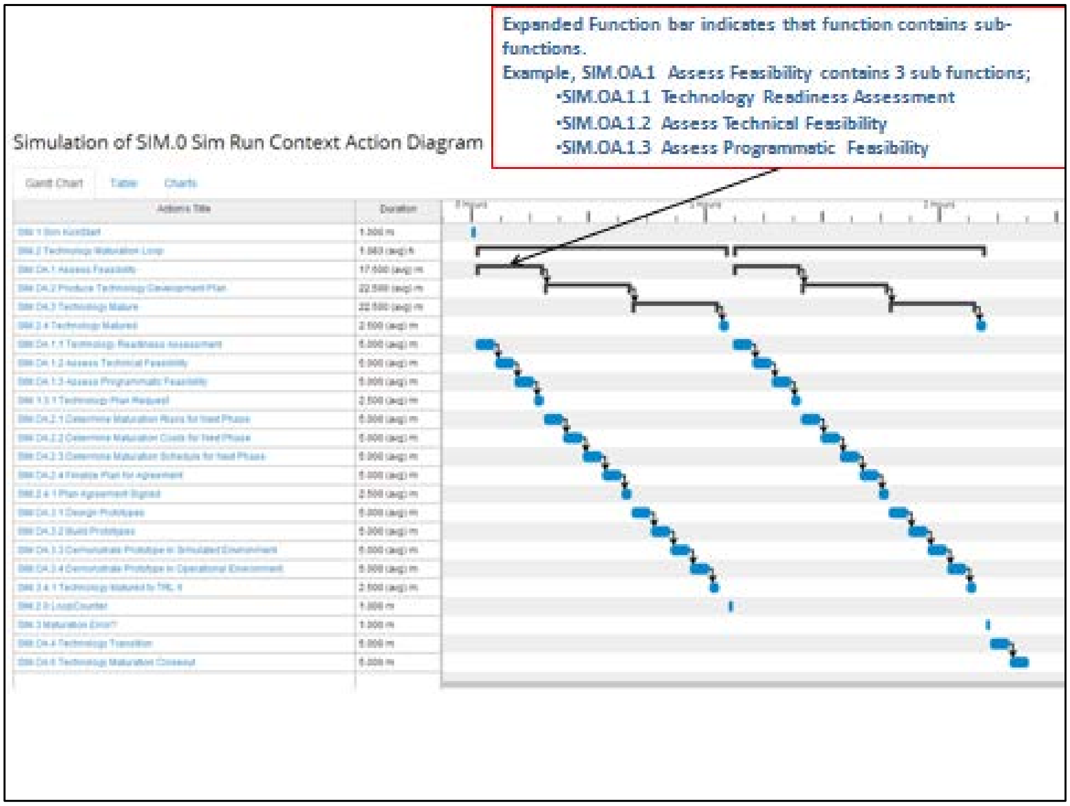


Figure 64. Simulation Gant Chart Sub-function Timeline Explanation

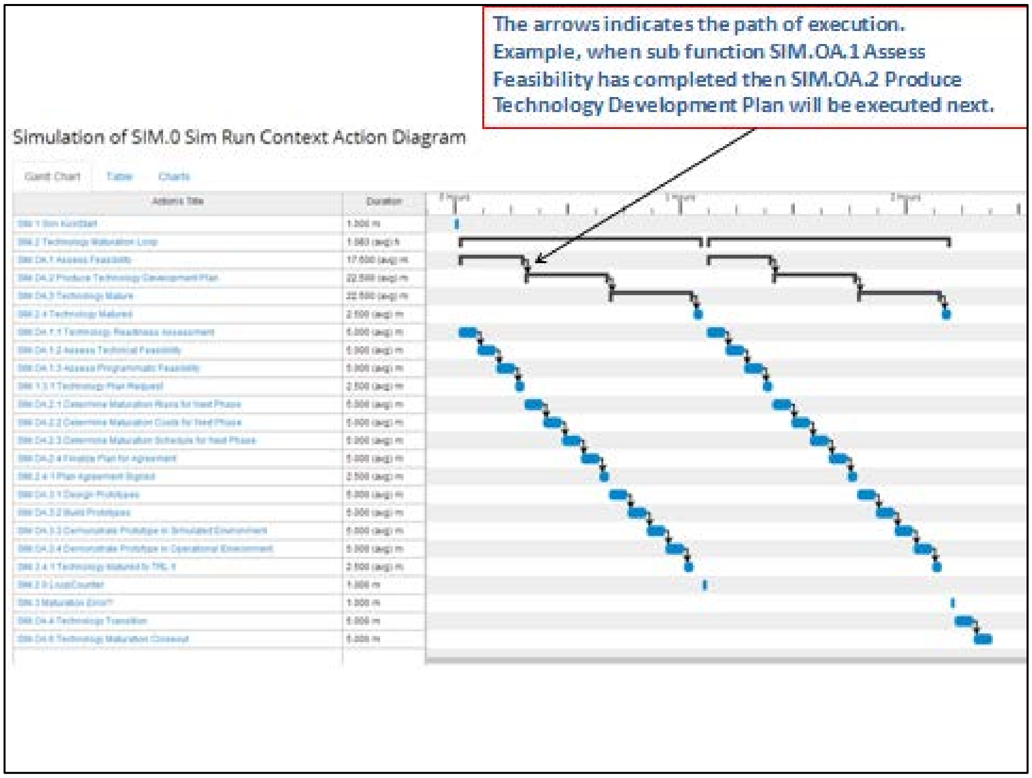


Figure 65. Simulation Gant Chart Timeline Path Explanation

LIST OF REFERENCES

3SL. 2014. "Extended Functional Flow Block Diagram." Accessed February 14 2014. https://www.threesl.com/pages/reference/diagrams/extended-functional-flow-block-diagram.php.

Acosta, Jacob, Scot Hoesly, Scott Huseth, Steven Krider, Jeremy Lamb, Calvin Martin, Jorge Medina, Vince Medina, Michael Nguyen, Jaykant Patel, Kamaljit Rangi, Tim Schoen, Khai Trinh and Ron Willis. 2007. Architecting Joint Command and Control System of Systems Capability Certifications. Monterey: Naval Postgraduate School.

AcqNotes.com. 2014. "Exit/Entrance Criteria." Accessed February 27 2014. http://www.acqnotes.com/Acquisitions/Exit%20Criteria.html.

AMRDEC. 2014. "Customer Solutions." Accessed February 20 2014. http://www.amrdec.army.mil/amrdec/pif/our_process.html.

ASD [R&E]. 2011. Technology Readiness Assessment Guidance. Washington, D.C.: DOD.

Asimov, Morris. 1962. *Introduction to Design*. Englewood Cliffs: Prentice-Hall.

Azizian, Nazanin, Dr. Mazzuchi, Dr. Sarkani, and Dr. Rico. 2011. "A Model for Measuring the Correlation Between TRA and Enabling Engineering Activities, Cost, Schedule, and System Quality for U.S. DOD Acquisition Programs." *13th NDIA Systems Engineering Conference.* 4–22. San Diego: National Defense Industrial Association (NDIA).

Bahill, A. T., and B. Gissing. 1998. "Re-Evaluating Systems Engineering Concepts Using Systems Thinking." *IEEE Transaction on Systems, Man and Cybernetics, Part C: Applications and Reviews.* Vol. 28, No. 4. 516–527.

Beimborn, Edward A. 2003. *Systems Analysis* - What, Why and How. Milwaukee: College of Engineering and Applied Science, University of Wisconsin-Milwaukee.

Blanchard, Benjamin S., and Wolter J. Fabrycky. 2011. *Systems Engineering and Analysis*. Fifth Edition. Upper Saddle River, NJ: Prentice Hall.

Borowski, Samuel Mark. 2012. "Competitive Prototyping In The Department of Defense: Suggestions for a Better Approach." *The Limits of Competition in Defense Acquisition, Defense Acquisition University Research Symposium.* 10–50. Fort Belvoir: Defense Acquisition University.

Buede, Dennis M. 2009. *The Engineering Design of Systems*. Second Edition. Hoboken: Wiley.

Carter, Jim A. 1992. “Managing: To Succeed with Rapid Prototyping.” *Human Factors and Ergonomics Society Annual Meeting.* 404–408.

Cate, Devin L. 1997. Where Have All the Prototypes Gone? The Failure of the Prototyping Initiatives of the 1990s. Air Command and Staff College.

Colquhoun, G.J., R.W. Baines, and Roger Crossley. 1993. “A State of the Art Review of IDEF0 .” *International Journal of Computer Integrated Manufacturing.* 252–264.

Copeland, Edward J, Dr. Thomas Holzer, Dr. Timothy Eveleigh, and Dr. Shahryar Sarkani. 2013. “Effects of System Prototype Demonstrations on DOD Weapon Systems Development.” *NDIA 16th Annual Systems Engineering Conference.* 4–17. Washington, D.C.: George Washington University.

Dahmann, Dr. Judith. 2010. “Early Systems Engineering.” *INCOSE Webinar Series.* Accessed February 7, 2014. https://www.incose.org/newsevents/events/details.aspx?id=94.

Dahmann, Judith Dr., and Aumber Bhatti. 2008. “Applying Business Process Modeling to Develop Systems Engineering Guidance for New DOD Acquisition Regulations.” Paper presented at the NDIA Systems Engineering Conference. 3–17. San Diego: The MITRE Corporation.

DARPA. 2013. Driving Technological Surprise: DARPA’s Mission in a Changing World. Washington, D.C.: DOD.

DAU. 2012. “Glossary Defense Acquisition Acronyms And Terms.” Fifteenth Edition. Fort Belvoir: Defense Acquisition University Press.

Department of Defense. 2013. Defense Acquisition Guidebook. Washington, D.C.: DOD.

Department of Defense. 2006. Risk Management Guide for DOD Acquisition. Sixth Edition. Washington, D.C.: DOD.

Department of the Navy. 2004. Naval Systems Engineering Guide. Washington, D.C.: Naval Systems Engineering Steering Group.

Dictionary.com. 2014. “Prototype.” Accessed March 3 2014. http://dictionary.reference.com/browse/prototypeDictionary.com.

DOD CIO. 2010. “DOD Architecture Framework.” Accessed February 9 2014. http://dodcio.defense.gov/dodaf20.aspx.

DOD. 2013. Department of Defense Dictionary of Military and Associated Terms. Washington, D.C.: DOD.

DOD. 2003. “Extension to: A Guide to the Project Management Body of Knowledge (PMBOK Guide)” First Edition, Version 1.0. Fort Belvoir: Defense Acquisition University.

DOD. 2013. “Interim DOD Instruction 5000.02: Operation of the Defense Acquisition System.” Washington, D.C.: DOD.

DOD. 2006. Risk Management Guide for DOD Acquisition Sixth Edition. Washington, D.C.: DOD.

DOD. 2013. Systems Management College. Systems Engineering Fundamentals. Washington, D.C.: DOD.

Drezner, Jefferey A., and Meilinda Huang. 2009. On Protoyping: Lessons from RAND Research. Santa Monica: RAND.

Drezner, Jeffrey A. 1993. The Nature and Role of Prototyping in Weapon System Development. Santa Monica: National Defense Research Institute, RAND.

Drezner, Jeffrey A. 1992. The Nature and Role of Prototyping in Weapon System Development. Santa Monica: National Defense Research Institute, RAND.

Ellis, Mike, and Jeff Craver. 2006. “Technology Program Management Model.” *National Defense Industrial Association.* 6–39. Arlington: Space and Missile Defense Technical Center.

Erwin, Sandra I. 2013. “Defense Technologists Advocate ‘Early Prototyping’ of Future Weapons.” National Defense Magazine Blog, April 23. Accessed November 8 2013. http://www.nationaldefensemagazine.org/blog/Lists/Posts/Post.aspx?ID=1123.

Evans, Bruce. 2000. The Requirements Engineering Process for Custom and COTS Based Systems. Lockheed Martin, Management and Data Systems.

Facktor, Debra, and John Colombi. 2012. Expedited Systems Engineering for Rapid Capability and Urgent Needs. Hoboken: Systems Engineering Research Center.

The Free Dictionary. “Design.” 2014. Accessed February 2, 2014. http://www.thefreedictionary.com/design.

The Free Dictionary. “Environments.” 2014. Accessed February 2, 2014. http://www.thefreedictionary.com/environments.

The Free Dictionary. “Full-scale.” 2014. Accessed February 2, 2014. http://www.thefreedictionary.com/full-scale.

The Free Dictionary. “Subject Matter Expert.” 2014. Accessed February 2, 2014. http://encyclopedia.thefreedictionary.com/Subject+Matter+Expert.

The Free Dictionary. “Termination.” 2014. Accessed February 2, 2014. http://www.thefreedictionary.com/termination.

Gallagher, Sean. 2012. How to blow $6 billion on a tech project. Accessed April 7 2012. http://arstechnica.com/information-technology/2012/06/how-to-blow-6-billion-on-a-tech-project/.

GAO. 2008. “2009 Is a Critical Juncture for the Army’s Future Combat System.” *Defense Acquistions.* Washington, D.C.: GAO.

GAO.1999. Best Practices: Better Management of Technology Development Can Improve Weapon System Outcomes. Washington, D.C.: GAO.

GAO. 2006. Best Practices: Stronger Practices Needed to Improve DOD Technology Transition Processes. Washington, D.C.: GAO.

GAO. 2011. Defense Acquisitions: Application of Lessons Learned and Best Practices in the Presidential Helicopter Program. Washington, D.C.: GAO.

GAO. 2013. Defense Acquisitions: Assessments of Selected Weapon Programs. Washington, D.C.: GAO.

GAO. 2006. Defense Acquisitions: Improved Business Case Is Needed for Future Combat System’s Successful Outcome. Washington, D.C.: GAO.

GAO. 2012. Urgent Warfighter Needs: Opportunities Exist to Expedite Development and Fielding of Joint Capabilities.” Washington, D.C.: GAO.

Glinz, Martin. 2007. “On Non-Functional Requirements.” Paper presented at the annual meeting of the IEEE International Requirements Engineering Conference. 21–26. Delhi: IEEE Computer Society.

Gordon, Mark. 2008. “The Need for Manufacturing Innovation and Readiness.” Paper presented at the annual NDIA Science and Engineering Technology Conference. 3–12. Charleston, South Carolina, April,15-17.

Goshorn, Deborah E. 2013. “Army Capabilities Determination Process.” Presented at the SE3250 Capabilities Engineering Coursework Module #2. 5. Monterey: NPS.

Haller, Norman M. 2013. Assessment to Enhance Air Force and Department of Defense Prototyping for the New Defense Strategy. Washington, D.C.: The National Academies Press.

Houde, Stephanie and Hill, Charles. 1997. *What do Prototypes Prototype?* In Handbook of Human-Computer Interaction, edited by M. Helander,T. Landauer, and P. Prabhu. Amsterdam: Elsevier Science.

Hoeferkamp, Rich, and Mike Zsak. 2007. "Risk Management In Systems Engineering." Presented at the DAU Hot Topics Forum, June 20.

Hoffman, Michael. 2011. "JTRS GMR Cost Rise Triggers Nunn-McCurdy Breach." Defense News, May 18. Accessed April 7 2014. http://www.defensenews.com/article/20110518/DEFSECT02/105180308/JTRS-GMR-Cost-Rise-Triggers-Nunn-McCurdy-Breach.

INCOSE. 2010. Systems Engineering Handbook : A Guide For System Life Cycle Processes And Activities. San Diego: INCOSE, SE Handbook Working Group.

Johnson, Dale. 2006. *Manned/Unmanned Common Architecture Program.* Presented at Aviation Applied Technology Directorate Status Meeting, Ft. Eustis, Virginia.

JROC. 2012. Manual for The Operation of the Joint Capabilities Integration. Washington, D.C.: DOD.

Keeney, Ralph L. 1992. *Value Focused Thinking, A Path to Creative Decisionmaking*. Cambridge: Harvard University Press.

Kendall, Frank. 2011. "DOD Letter To Congress On JTRS GMR Termination." Letter to the House Armed Services Committee, October 13.

Kirkwood, Craig W. 1997. Strategic Decision Making, Multiobjective Decision Analysis with Spreadsheets. Belmont: Wadsworth Publishing Company.

Knowledge Based Systems. 1993. Integration Definition for Function Modeling (IDEF0). College Station: Knowledge Based Systems.

Kowal-Jurgens, Teresa. 2011. Six Common New Product Development Metrics. Accessed January 25 2013. http://www.globalnpsolutions.com/services/npd-resources/white-papers/6-common-npd-metrics/.

Kwinn, Brigette. 2012. " Value System Design." Presented at the SE3100 Fundamentals of System Engineering Coursework Module #4. Monterey: NPS.

Lai, Allen, William Millette, Doug Glandon, and Jeremy Royster. 2013. "Forward Resusitation Team Man-Portable Plasma Storage." Presented at the SI3400 Fundamentals of Eng Project Management Coursework. Monterey: NPS.

Lane, Jo Ann, Barry Boehm, Mark Bolas, Azad Madni, and Richard Turner. 2010. "Critical Success Factors for Rapid, Innovative Solutions." MS thesis, Hoboken: Stevens Institute of Technology.

Langford, Gary O. 2012. *Engineering Systems Integration*. Boca Raton: CRC Press.

Liou, Frank W. 2008. *Rapid Prototyping and Engineering Applications*. Boca Raton: CRC Press.

Long, David, and Zane Scott. 2011. A Primer for Model-Based Systems Engineering. Second Edition. Blacksburg: Vitech Corporation.

Merriam-Webster. 2014. "Characteristic." Accessed February 4 2014. http://www.merriam-webster.com/dictionary/characteristic.

Merriam-Webster. 2014. "Editor-in-Chief." Accessed February 4 2014. http://www.merriam-webster.com/dictionary/editor%20in%20chief.

Merriam-Webster. 2014. "Laboratory." Accessed February 4 2014. http://www.merriam-webster.com/dictionary/laboratory.

Merriam-Webster. 2014. "Prototype." Accessed January 16 2014. http://www.merriam-webster.com/dictionary/prototype.

Merriam-Webster. 2014. "Technology." Accessed February 4 2014. http://www.merriam-webster.com/dictionary/technology.

Morrow, Patrick K. 2011. *Analysis of Alternatives*. Fort Belvoir: Defense Acquisition University Press.

NASA. 2008. 2008 NASA Cost Estimating Handbook. Washington, D.C.: NASA.

NASA. 2007. NASA Systems Engineering Handbook. Washington, D.C.: NASA.

Oliver, David W., Timothy P. Kelliher, and James G. Keegan Jr. 1997. "Engineering Complex Systems.." First Edition. New York: McGraw-Hill Companies.

OUSD(AT&L) Systems and Software Engineering/Enterprise Development. 2006. "Risk Management Guide for DOD Acquisition." Sixth Edition, Version 1. Washington, D.C.: DOD.

Oxford University Press. 2014. "Statuatory." Accessed February 2, 2014. http://www.oxforddictionaries.com/us/definition/american_english/statutory?q=Statutory.

Packard, David, 1986. “A Quest for Excellence: Final Report to the President.” The President’s Blue Ribbon Commission on Defense Management. Washington, D.C.

Pawlowski, Tom, Paul C. Barr, and Steven J. Ring. 2004. Applying Executable Architectures to Support Dynamic Analysis of C2 Systems. Fort Leavenworth: Mitre Corporation.

Plato, John J. 1995. “Prototyping: Proceed with Caution.” Information Systems Management. Vol. 12, Iss. 4: 69–71.

Pressman, Roger S. 2010. *Software Engineering: A Practitioner’s Approach*. Seventh Edition. New York: McGraw-Hill Companies.

Reed, John. 2012. “Oh Canada: The Fate of the Marines’ VH-71 Fleet.” Defense Tech, April 19. Accessed April 5 2014. http://defensetech.org/2012/04/19/oh-canada-the-fate-of-the-marines-vh-71-fleet/.

Sage, Andrew P., and William B. Rouse. 2009. *Handbook of Systems Engineering and Management*. Hoboken, NJ: John Wiley & Sons.

Scott, Zane. 2011. Model Based Systems Engineering. Blacksburg: Vitech Corporation.

SPEC. 2012. “Overview: What is Innoslate?” Accessed February 8 2013. https://innoslate.com/overview/.

Sullivan, Michael J. 2013. “Weapons Acquisition Reform: Reform Act is Helping DOD Acquisition Programs Reduce Risk, but Implementation Challenges Remain.” Paper presented at the annual Acquisition Research Symposium: Acquisition Portfolio Trends. 7, 8–13. Monterey, California, May 15–16.

TRADOC. 2011. Regulation 71–20: Concept Development, Capabilities Determination, and Capabilities Integration. Fort Monroe: Department of Army.

TRADOC. n.d. Action Officer: Staff Writing. Fort Monroe: Department of Army.

Tripp, Steven D., and Barbara Bichelmeyer. 1990. “Rapid Prototyping: An Alternative Instructional Design Strategy.” Educational Technology Research and Development, Vol. 38, Iss. 1: 31–44.

USD [AT&L]. 2012. “Scientific and Engineering Integrity.” DODI 3200.20. Washington, D.C.: DOD.

USD [AT&L]. 2013. “Operation of the Defense Acquisition System.” Interim DODI 5000.02. Washington, D.C.: DOD.

Valerdi, Ricardo, and Ron J. Kohl. 2004. “An Approach to Technology Risk Management.” Paper presented at the Engineering Systems Division Symposium, Cambridge, Massachusetts, March 29–31.

Warfel, Todd Zaki. 2009. *Prototyping: A Practitioner’s Guide.* Brooklyn: Rosenfeld Media.

Weinberg, Randy S. 1991. “Prototyping and the Systems Development Life Cycle.” Journal of Information Systems Management 8: 47–53.

Wiegers, Karl E. 1999. “Writing Quality Requirements.” Software Development, May.

Wilbur, Lee, and Allan Steinhardt. 2012. *Rapid Prototyping: The Agile Creation of Solutions for Modern Defense & Intelligence*. McLean: Booz Allen Hamilton.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California